AD NUMBER
AD911395
NEW LIMITATION CHANGE
TO Approved for public release, distribution unlimited
FROM Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; FEB 1973. Other requests shall be referred to Naval Ship Systems Command, Attn: PMS-302-4, Washington, DC 20362.
AUTHORITY
USNSSC ltr, 29 Mar 1974

AD 911395

Contract N00024-70-C-1274
Project Serial No. SF11-121-106
Task 8523
TRACOR Project 035 001 01
Document Number T73-AU-9519-U

FINAL REPORT

COMPUTER AIDED PROCESSING FOR ACTIVE AND PASSIVE SONAR SYSTEMS

by

H. A. Reeder
and
D. W. Hamm

Submitted to

Commander
Naval Ship Systems Command
Department of the Navy
Washington, D. C. 20362

Attention Mr. John Neely, Code 302-4

14 February 1973

TRACOR

6500 Tracor Lane, Austin, Texas 78721, AC 512/926-2800

TRACOR 6500 TRACOR LANE, AUSTIN, TEXAS 78721

Contract N00024-70-C-1274
Project Serial No. SF11-121-106
Task 8523
TRACOR Project 035 001 01
Document Number T73-AU-9519-U

FINAL REPORT

COMPUTER AIDED PROCESSING FOR ACTIVE AND PASSIVE SONAR SYSTEMS

by
H. A. Reeder
and
D. W. Hamm

Submitted to

Commander
Naval Ship Systems Command
Department of the Navy
Washington, D. C. 20362

Attention Mr. John Neely, Code 302-4

14 February 1973

Approved by:

W. C. Moyer
Division Vice President

Prepared by:

H. A. Reeder
Project Director

TRACOR 6500 TRACOR LANE, AUSTIN, TEXAS 78721

ACKNOWLEDGEMENTS

We wish to express our appreciation to:

1. Mr. J. J. Dow, Dr. B. M. Brown, and Dr. W. C. Moyer for many fruitful discussions and contributions to this report.

2. Mr. W. B. Butler for his assistance in the conduction of the display observer study.

3. Mr. Pat Pitt and Dr. Charles Wood of the Applied Research Laboratory of The University of Texas for their cooperation in furnishing data for the observer study.

4. Mr. John Neely, Mr. Matt Giwer, and Mr. Ken Buske for sponsoring this study and providing pertinent suggestions.

We further wish to express our appreciation to Carolyn Atkinson, Betty Fischer, Martha Foehner, Pat Keller, and June Vogel for preparing the text and figures for this report.

TRACOR 6500 TRACOR LANE, AUSTIN, TEXAS 78721

ABSTRACT

A general framework for a computer system that accepts and analyzes the vast quantity of data generated by a modern sonar suite has been developed. The output of this computer system is an array of alerting functions that measure the likelihood that a given coordinate vector is the location of a target. In particular, the framework combines the output of active high- and low-Doppler sonar processors and wideband and narrowband passive processors. The active high- and low-Doppler processor portion was developed and tested during this study. Performance tests using simulated data established that the combined active processor gave better performance than each individual processor and its single output channel gave more uniform performance over variations in target Doppler than was available with the two separate channels. An observer test was conducted using ARL processed AN/SQS-23 recorded sea data with injected target signals.

TRACOR 6500 TRACOR LANE, AUSTIN, TEXAS 78721

TABLE OF CONTENTS

TABLE OF CONTENTS--Continued

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS--Continued

LIST OF TABLES

1.0 INTRODUCTION

Two of the major problems faced by modern sonar systems are the consolidation and presentation of a potentially large quantity of data and the ability of the sonar operator to effectively assimilate and process these data. Potentially, a modern sonar suite would be capable of delivering thousands of channels of information to the operator who in turn, even in an alert state, cannot process all of this information. Moreover, operators do not typically perform in an alert manner when required to search for extended periods of time, especially when the incidence of contacts is low. The result of this is that submarines may go undetected for longer than necessary and when detected the resultant time available for classification and tracking is reduced--possibly to an extent that seriously degrade the ASW system's performance. Our approach to the solution of this problem is to develop a computer processing system which can handle the vast quantity of data and in so doing operate in a near optimum manner by virtue of a ping-to-ping integration and tracking algorithm.

Work under this contract has produced a general framework for a computer system that will accept and analyze this vast quantity of data. The output of this system will be an array of alerting function values that measure the likelihood that a target occupies each of coordinate positions within the search volume of the sonar. Information concerning the target track is stored in the computer bank. Toward implementing the above framework, the present study has developed a computer processor which analyzes the outputs of two active processors, high- and low-Doppler, and combines the results into a single output channel which can be used for decision purposes. Also an observer study to measure the performance of a low-Doppler SLR processor using sea data was carried out. An SLR processor for a wideband passive sonar was developed.

The processing is based on Sequential Likelihood Ratio (SLR) procedures that have been investigated previously. Those results are summarized in Appendix A. Briefly, the SLR processor combines a statistical decision test (Wald's Sequential Probability Ratio Test) and a basic tracking program. The tracking program selects target tracks that have motion consistent with that of a submarine and the SLR test is used to decide whether the track should be rejected or retained, and, if retained, possibly displayed. The testing procedure operates much like an alert operator but without the variability of an operator who is of course susceptible to fatigue, subjectivity, boredom, poor training and a host of other deterrents to ideal, time-invariant detection performance. More important than this perhaps is the fact that the information handling capacity of this computer process is far in excess of that of the operator and is also subject to expansion as computer technology improves whereas the capacity of the operator is unlikely to be expanded or improved by any significant amount in spite of advances in display technology.

2.0 SUMMARY OF RESULTS

Several major items have been accomplished during this study. These were:

1. Development of a general framework for combining multiple sonar receiver outputs to form alerting functions.

2. Extension of sequential likelihood ratio (SLR) processing to three dimensions—range, bearing, and Doppler.

3. Combination of two active sonar processed channel outputs—low- and high-Doppler—in an SLR algorithm.

4. Implementation of a version of the SLR processor on The University of Texas Applied Research Laboratory's (ARL) CDC 3200 digital computer

5. Development of a mathematical model to calculate the probabilities of clutter and detection in the SLR processor.

6. Conducting an observer-display study using ARL data to test the SLR processor with sea data.

7. Development of an SLR processor for the output of a wideband passive sonar.

The primary steps for combining multiple sonar receiver outputs include SLR processing of each individual signal processing channel, weighting the results depending on expected processor performance, combining channels that have overlapping performance envelopes, e.g., wideband and narrowband passive processors, and choosing the maximum output among the individual

and combined processors. Provisions must be made for dimensionality mismatch and varying resolution cell sizes. The multireceiver processor is described in detail in Section 3.0.

The SLR multireceiver processor was implemented to simultaneously process multiple active sonar receiver outputs. In particular, a sonar system consisting of both low-Doppler (FM replica correlator) and high-Doppler (CW comb-filter bank) receivers was assumed. The previously developed SLR tracking and detection algorithm which employed the dimensions of range and bearing has been generalized to include the Doppler dimension which is available at the output of the high-Doppler receiver. In implementing the high-Doppler SLR it was found that retaining only the maximum amplitude sample associated with the comb-filter bank for any given resolution cell gave almost equivalent detection performance and required 64% less computer space than retaining all significant samples. The development of the two single channel SLR processors is discussed in Section 4.0 of a previous report.*

Based on the general framework, a processor which combines the two active processor outputs was developed and tested. It was found that the combined SLR gave better performance than either single channel SLR and that the performance of the combined SLR was more consistent across the range of Doppler shifts than was each individual processor. This was true even for the range of Doppler shifts where the performance of both processors was degraded. The details of this study are given in Section 5.0 of the above mentioned report.

The output of ARL's digital sonar was processed with an SLR processor developed for the ARL CDC 3200 computer. This

*Reeder, H. A., "Simultaneous Likelihood Ratio Processing for Two Active Receivers," TRACOR Document T71-AU-9594-U, Vol. I, 23 August 1971.

version operates on amplitude samples indexed by range and bearing. In order to efficiently use the smaller, less sophisticated machine, this new SLR processor differs substantially in details from the TRACOR UNIVAC 1108 computer version. However, the final results of the two programs, operating on the same data, are identical. This program is discussed in Section 4.0.

The mathematical model of the SLR processor clutter and detection probabilities makes possible the easy and economical study of the effects of parameter changes in the processor. The model and one such study are discussed in Section 4.3 of TRACOR Document T71-AU-9594-U. The parameter study indicates that the greatest gains in individual SLR processor performance may be accomplished by making the SLR tracking algorithm more selective--i.e., reducing the ping-to-ping tracking errors.

The ARL data used in the observer display study were recorded AN/SQS-23 stave data with injected target signals. These data were digitally beamformed by ARL, envelope detected, and integrated. Local peaks in bearing and range were retained and a mean and standard deviation of the local noise field calculated. These latter statistics were used to normalize the peak amplitude values. It was found that some data runs exhibited noise spokes that interfered with SLR processing. Ways to partly suppress these spokes were implemented. The first twelve injected targets simulated straight line motion and the signal-to-noise ratio was increased 2 dB every five pings. The last run was a maneuvering target with constant signal-to-noise ratio. These data were analyzed by the SLR processor implemented on the ARL CDC 3200 computer. The SLR and non-SLR processed data were analyzed and compared, first analytically, then with observer responses. It was found analytically that the SLR processed data crossed decision thresholds sooner and therefore at lower signal-to-noise ratios. However, the observer responses showed no

statistically significant differences between the SLR and non-SLR processed data for the limited data base available for the test.

One possible reason for this is the fact that the signal-to-noise ratio increased periodically and the probability of marking the display for both SLR and non-SLR processed data went from low to high in a short time period. This "bang-bang" effect masks the ability of the SLR processor to integrate and enhance small signal-to-noise ratio signals. Also, the display study utilized alerted observers who were not subjected to the fatigue of long watches with low incidence of a target. This factor would be difficult to reproduce in the laboratory. Finally, the data base consisted of only 13 independent runs. With such a small data base, only gross differences will show statistically significant differences. The results of this study are given in Section 5.0.

The SLR processor was adapted to the output of a wideband passive receiver. The results for simulated data show an improvement in the form of reduced input signal-to-noise ratio to achieve 0.5 probability of detection at a set false alarm probability. The results of this study are reported separately.*

*Reeder, H. A., "Computer Aided Detection for Wideband Passive Sonar Systems," TRACOR Document T73-AU-9520-U, 14 February 1973.

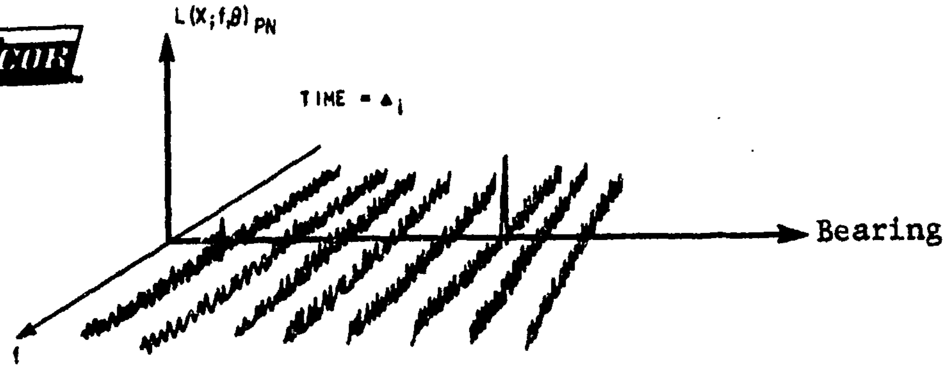
3.0 FRAMEWORK FOR INCLUSION OF MULTIPLE SONAR RECEIVER OUTPUTS IN SLR PROCESSING

3.1 Formation of Multireceiver Joint Likelihood Ratios--Alerting Functions

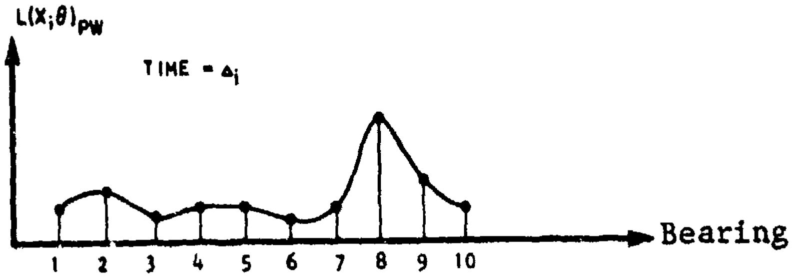
The objective of this section is to develop a multichannel alerting algorithm for various input channels as well as high- and low-Doppler active search receiver outputs. Sequential likelihood ratio tracking of active receiver outputs is the subject of a present contract. It is assumed that each separate channel has been subjected to a sequential likelihood ratio process that yields a likelihood ratio for each resolution cell. For purposes of illustration four processors will be considered: a low-Doppler processor whose output is indexed by range and bearing; a high-Doppler processor, indexed by range, bearing, and Doppler shift; a wideband passive processor, indexed by bearing; and a narrowband passive receiver, indexed by bearing and line frequency. These four processors demonstrate a variety of situations inherent in combining likelihood ratios with different dimensionality and resolution cell sizes.

When all of the SLR ping-to-ping tracking is accomplished on each receiver output, there will exist the information shown in Fig. 3-1. For convenience, this figure shows the processed data in a continuous and unthresholded form, although in reality, each output is sampled and thresholded so that the actual quantity of data will be less than that shown. The task before us now is to adopt a method for combining these outputs to form a measure of the likelihood that a target occupies a given range-bearing cell. This will be approached by first combining the two processed active outputs, then (conceptually) combining the two passive system outputs and finally (again conceptually) combining the

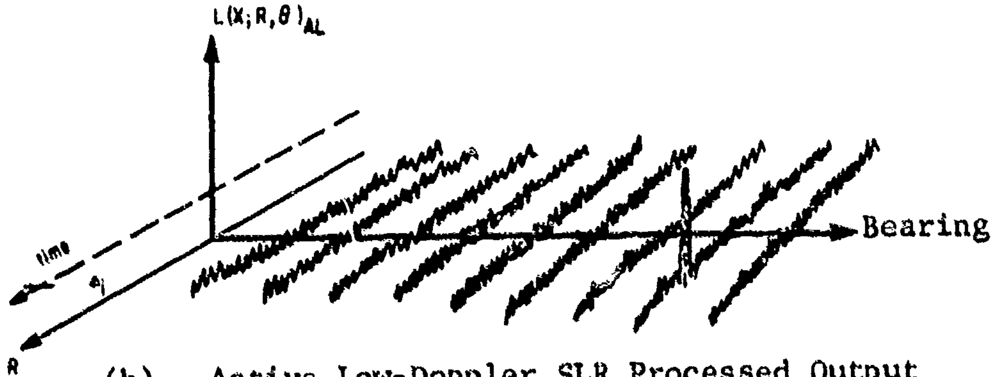
TRACOR



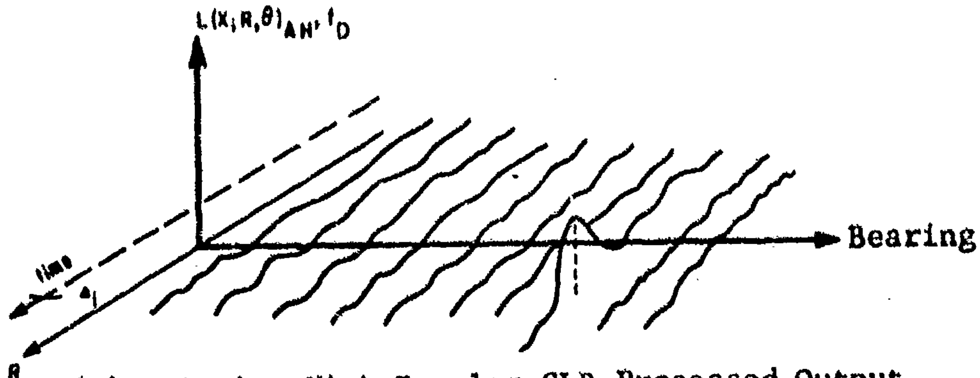
(d) - Passive Narrowband SLR Processed Output



(c) - Passive Wideband SLR Processed Output



(b) - Active Low-Doppler SLR Processed Output



(a) - Active High-Doppler SLR Processed Output

FIG. 3-1 - REPRESENTATIONS OF THE SLR PROCESSED ACTIVE AND PASSIVE RECEIVER OUTPUTS

joint active and passive outputs. The general idea here is to first combine those processors that are most similar in processed data form and then combine the results of these combinations.

In combining the outputs of two or more processors we face problems related to different resolution cell size and dimension mismatch. For example, in the active high- and low-Doppler receivers we have tracks developing in the range, bearing and Doppler dimensions of the former while tracks in the latter develop in only the range and bearing dimensions. In addition, it is possible to have a range cell size mismatch between these two processor outputs since resolution in the low-Doppler receiver is determined by the coded pulse bandwidth while range resolution in the high-Doppler receiver is determined by the duration of a long CW pulse. These differences are depicted in Figs. 3-1a and 3-1b where in the case of the high-Doppler output the ordinate is described by a pair of numbers, one giving the track likelihood ratio, the other giving the associated Doppler. Clearly, for any given range-bearing cell in the high-Doppler system there can be more than one (likelihood ratio, Doppler) pair since tracking is occurring in the Doppler dimension. To reduce the dimensionality to the range-bearing dimensions, the computer will retain only the maximum likelihood ratio (and its associated Doppler) in each range-bearing cell. This eliminates the problem of equalizing the dimensionality between the two active systems.

3.1.1 Combination of Low- and High-Doppler Active System Outputs - The next problem is one of mismatch between resolution cell size. Figure 3-2 shows a representation of the outputs of the SLR processed high- and low-Doppler active systems from preformed beams steered in the same direction. The outputs have

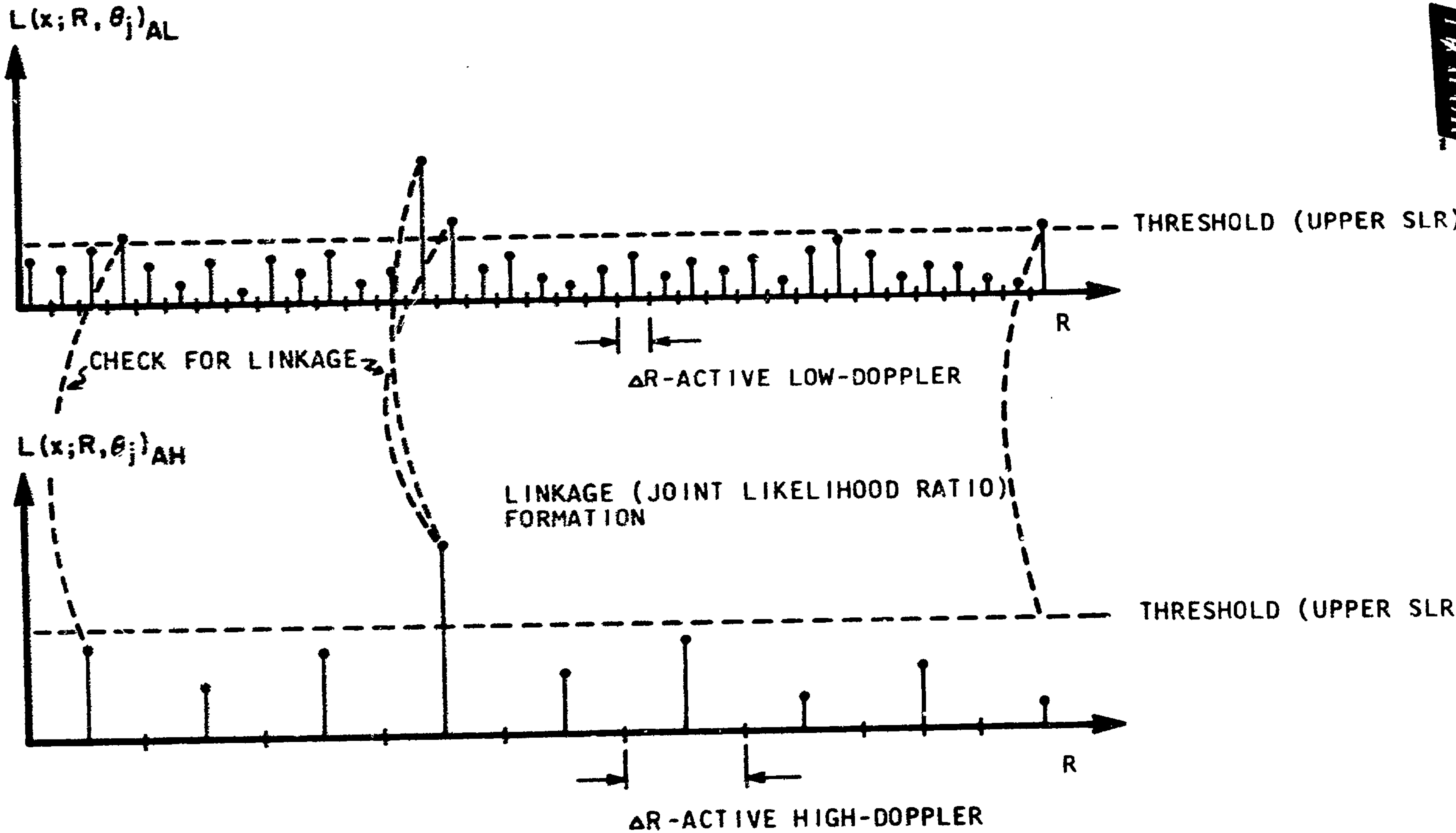


FIG. 3-2 - REPRESENTATIONS OF THE JOINT LIKELIHOOD RATIO FORMATION BETWEEN HIGH- AND LOW-DOPPLER SLR PROCESSED OUTPUT DATA

been time delayed as necessary so that each time resolution cell covers the same range. The output from the low-Doppler SLR processor has been serially OR-ed over a range gate corresponding to the approximate length of a target or a display resolution cell. This is done because eventually the output of the combined SLR must be matched to a display for presentation and the serial OR at this point reduces subsequent processing with almost no degradation in performance. In any case the low- and high-Doppler outputs are combined as shown in Fig. 3-2. This figure shows that in the low-Doppler system three threshold exceedings have occurred during the echo cycle shown. Each of these events precipitates a check for a threshold exceeding event in the high-Doppler SLR processed system output within high-Doppler range resolution cells which encompass the range cell containing the low-Doppler event. If threshold exceedings do not occur on both system outputs within a common high-Doppler range cell, then no track linkage is made during the echo cycle under consideration. In this way the SLR processed data from both active receivers is combined into a common measure of the likelihood of target.

3.1.2 Combination of Wideband and Narrowband Passive System Outputs - It is desirable to find a single quantity which represents both channels of passive information, or, more generally, a quantity that represents the likelihood of a target being on a particular bearing only. The first problem faced in finding a joint measure of passive information is a dimensionality mismatch. For any given time the narrowband information is indexed by frequency and bearing while the wideband information is indexed by bearing only. Probably, the best way to approach this problem is to adopt the same procedure as in the active case and reduce the dimensionality of the narrowband data. This may be done by a Max-OR or summing process. The summing approach would consist of

making threshold tests on the spectral outputs of each beam and then summing those frequency channel outputs which exceed a threshold.

Once the dimensionality has been matched, a combination process such as the active case may be carried out. This involves adding log likelihood ratios that correspond to the same bearing resolution cell and that exceed certain thresholds. The results are joint log likelihood ratios that form a measure that there is a passively-detected target associated with that bearing resolution cell.

3.1.3 Combination of Active and Passive SLR Processed Data - It is assumed that each active and passive beam will have a joint log likelihood ratio as a function of time associated with it. The task is now to combine active and passive information. Since each information channel is obtained in a different way and in separate frequency bands, it is reasonable to assume tentatively that they are statistically independent; hence, the joint active-passive log likelihood ratio is just the sum of the two individual log likelihood ratios. The essential problem in this case is the dimensionality mismatch mentioned previously. The passive information is indexed by bearing and time only while the active is indexed by range, bearing and time.

Recall from Fig. 3-1 that for the passive systems we have log likelihood ratios indexed by bearing and time. This means that when the joint log likelihood ratios are formed, we will have for the passive receivers a single measure of the likelihood of target for each resolvable bearing and at each instant of time. Similarly, for the active systems we will have a set of joint log likelihood ratios indexed by range, bearing, and time or ping

number. These arrays of numbers are shown diagrammatically in Fig. 3-3. This figure shows the output of the joint SLR processed passive data which exists at time Δ_i, as well as the joint SLR processed active data during a ping cycle which exists over a time period that encompasses the time Δ_i. Since no range information is currently available from the passive search system, the selected active joint log likelihood ratios will be enhanced by summing them with the last available and relevant passive log likelihood ratios that occur on the same bearing. By "selected" we mean that a threshold test would be applied to the active data before linking to it the thresholded passive data. The levels at which these thresholds will ultimately be set will depend upon the available computer capacity. That is, ideally it would be desirable to perform no thresholding until all of the joint active and passive log likelihood ratios have been formed. However, from a realistic viewpoint there must be the capability for reducing the amount of data stored in the shipboard computer.

The computer algorithm which will perform sequential likelihood ratio tracking and joint log likelihood ratio formation on the outputs of active high- and low-Doppler and passive narrowband and wideband outputs is shown in block diagram form in Fig. 3-4.

This figure shows each of the active and passive SLR processed outputs being applied to threshold circuits and then to weighting circuits. The purpose of the threshold circuits is primarily to control computer loading. This will be accomplished in the following manner. The outputs of each of the SLR processors shown in Fig. 3-4 will be unthresholded tracks and thus will contain numerous spurious noise tracks. Ideally, we would prefer to defer any decision with regard to threshold as late in the processing as possible so that low signal-to-noise ratio tracks will be enhanced as much as possible by the process of joint tracking and

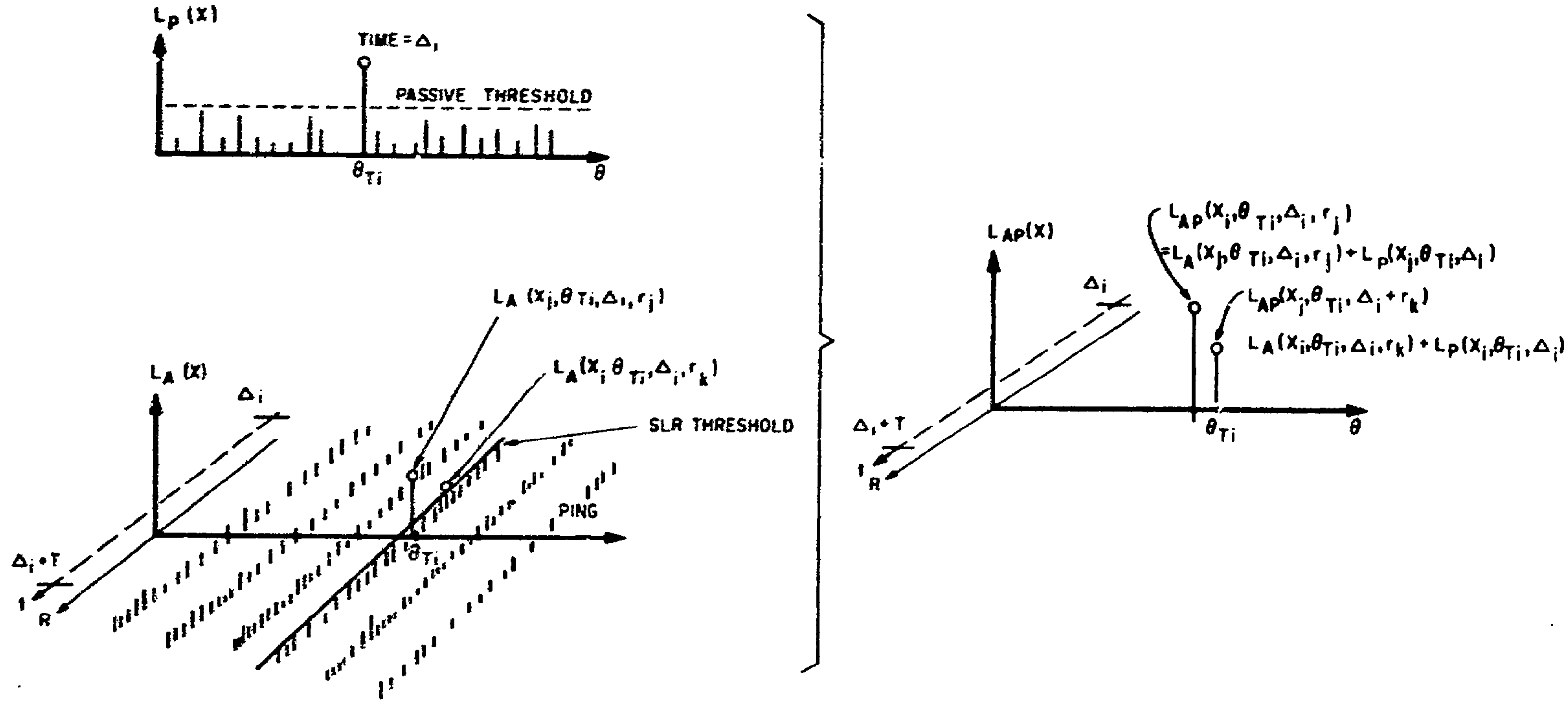


FIG. 3-3 DIAGRAMMATIC REPRESENTATION OF THE JOINT ACTIVE-PASSIVE LOG LIKELIHOOD RATIO FORMATION PROCESS

M 11/4/70 86-62-4296 JO

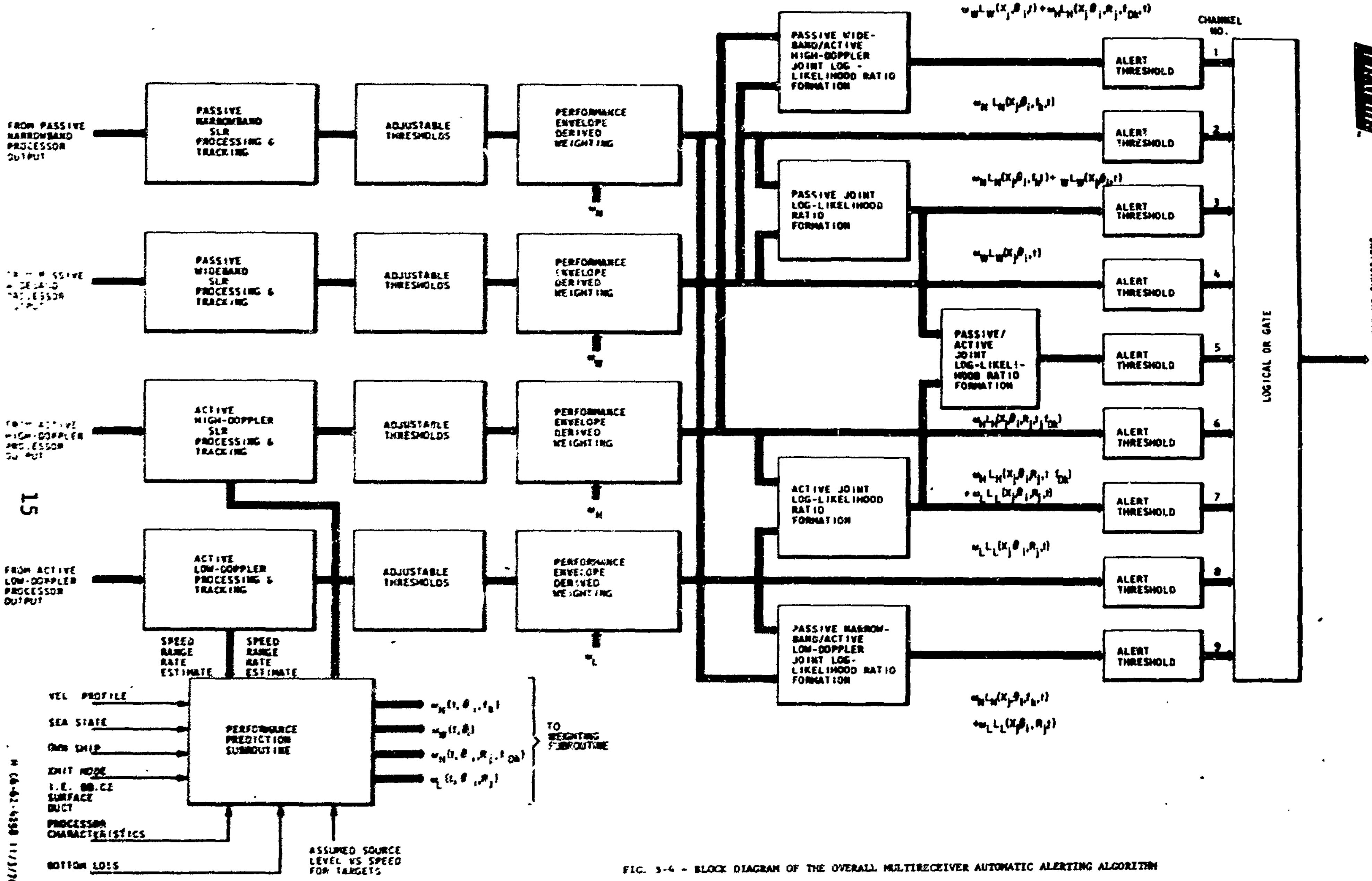


FIG. 3-4 - BLOCK DIAGRAM OF THE OVERALL MULTIRECEIVER AUTOMATIC ALERTING ALGORITHM

H C6-62-4358 11/3/70 RN:JD

likelihood ratio formation. However, it will be necessary to provide a means of controlling the number of tracks so that the computer storage capacity is not exceeded. This then is the function of the adjustable thresholds shown in Fig. 3-4.

The thresholded data are then passed to circuits which weight the log likelihood ratios according to a predicted performance envelope. The weighting of the SLR processed output of each sonar receiver is necessary. When one can predict that a certain processor will not perform beyond a certain range under a specific set of conditions, then it is obvious that the inclusion of that processor's output in the final joint log likelihood ratio will only serve to degrade otherwise potentially sound tracks. As an example of this use of weightings consider the situation where the speed of a given track in the active systems is estimated to be near zero. There are other conditions such as own ship speed, sea state, and propagation loss which when combined with sufficiently low target speed would result in a very low probability of threshold crossing in the passive wideband receiver at all but very small target ranges. Such a prediction will of course require that one assume some radiated spectrum level versus speed function for the potential target. This may be accomplished by considering those enemy submarines of interest which exhibit the highest radiated noise level as a function of speed and obtaining an average relationship for these targets to be used as input to the performance prediction subroutine. This approach will result in a maximum average detection envelope and thus will result in weightings for the wideband passive system output which will ensure (statistically) that all targets of interest will be processed. When in error, as for example when a quieter submarine is actually present, the overall log likelihood ratios will suffer some degradation as a result of the conservative manner in which weightings are derived. However, the individual receiver log likelihood ratios will not be degraded.

These weights will, without any other data available, be either zero or one, and will be indexed by range and bearing angle. Fig. 3-5 shows an example of the weighting function for the wideband passive system as well as the manner in which it is derived. As shown in this figure, the probability of exceeding threshold as a function of range is used to determine some performance threshold and hence a maximum performance range R_{max}. This information is used to form the weighting function which acts, in fact, as a performance envelope gate for the data emerging from each channel of the multireceiver system.

At the output of the weighting circuits the four channels of data are combined not only into active and passive joint log likelihood ratios and a joint active/passive log likelihood ratio but also in various other ways. Namely, in addition to the joint log likelihood ratios just mentioned, the individual log likelihood ratios are preserved separately as well as combined into other joint functions. Specifically, the joint log likelihood ratios may be based on passive wideband and active high-Doppler outputs as well as passive narrowband and active low-Doppler outputs. In this way the system consists of nine different channels of output SLR data.

Consider for a moment the reasons for producing an output of this type. To begin with, if we could predict exactly for a given target and environment, the performance envelope of each of the four receivers and thus quite accurately accept or reject those receiver outputs which should or should not contribute constructively to the overall joint log likelihood ratio, then there would be no need for a multichannel output. To some extent we can do this--in particular according to the weighting mechanism described earlier. However, it is not possible to forecast a priori the exact performance envelope for every target type and environment

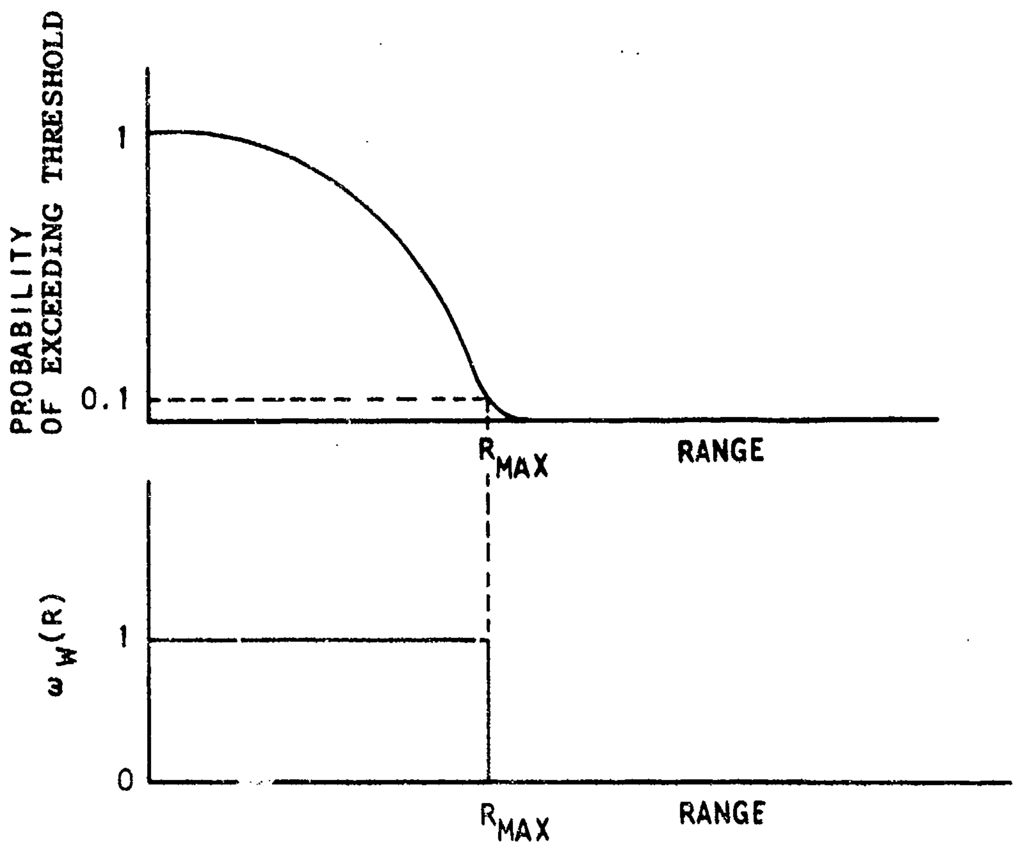


FIG. 3-5 - PERFORMANCE ENVELOPE AND RESULTANT WEIGHTING FUNCTION FOR THE WIDEBAND PASSIVE SYSTEM

and then select the proper envelope and weight. Since we have chosen to use performance envelopes based on best case characteristics (e.g., a noisy target) there will arise cases where a receiver has no chance of detecting the target even though the weight is equal to unity (e.g., a very quiet target). In this instance the wideband passive receiver will contribute essentially a noisy output to the joint likelihood ratio. In general there may very well be cases where one channel by itself could cause an alarm but when combined with other nonperforming, noisy channels of data, the chance to detect is lost. This possibility can be dealt with by allowing each channel to generate an alarm on its own.

Actually, the situation we have here is quite analogous to the problem of detecting a narrowband signal in a wideband noise background when the carrier frequency of the signal is not known a priori.

It is known that the optimum approach to detecting this signal is to design a bank of contiguous filters, each having a center frequency which the signal carrier frequency may conditionally take on.* Thus each channel of this system is optimum for detecting the signal under the condition that the carrier frequency of the signal is equal to the center frequency of the channel. The remaining part of the processing in this case is to select the filter output whose likelihood ratio is maximum. By analogy then we can view our multichannel system as consisting of several channels each of which is optimum under some condition which cannot be predetermined. That is to say, if the performance curves of each of the four input channels overlap for some set of

*E. J. Kelly, I. S. Reed, and W. L. Root, "Detection of Radar Echoes in Noise I." Journal of the Society of Industrial Applied Mathematics, pp 309-341, Vol. 8, No. 2, June 1960.

conditions, then the overall joint active/passive log likelihood ratio channel will be the optimum channel and will control the output of the OR-gate shown in Fig. 3-4. If on the other hand, the target is a high speed, bow aspect target such that these and other conditions render the active high-Doppler and passive wideband combination optimum, then this channel will dominate the OR-gate output and lead to alerting functions.

The price that is paid by taking this approach is of course an increased number of opportunities to false alarm which can be compensated for by increased thresholds which in turn leads to some decrease in detection capability. However, theoretical considerations indicate that there will be a net gain by this approach.

It will be recognized in Fig. 3-4 that not all combinations of the four receiver outputs are considered. There are in fact fifteen different combinations which could be formed from the original four channels of data. We have selected certain channels which appear reasonable. That is, we have the overall joint active/passive channel which will be optimum when all performance curves overlap. There are also the individual log likelihood ratio channels, one of which can produce an alarm should the other three channels be inoperable by virtue of constructed performance envelopes. The combination of both passive channels results in a detection channel which should be effective against torpedoes where active receiver performance is seriously degraded by small target strengths. The combined active channels are effective against a deep, quiet submarine running at a speed greater than zero knots but just below the cavitation inception speed. The combination of active high-Doppler and passive wideband is a reasonable choice for a high speed sub, while the combination of active low-Doppler and passive narrowband receiver outputs may be the optimum means of detecting a very low speed sub.

What must be established is the performance of different configurations of the automatic alerting system under a variety of input conditions which represent real world situations. This is necessary so that we can make an intelligent choice of a single configuration with respect to both detection performance and computer requirements.

4.0 SLR PROCESSOR FOR ARL's DIGITAL SONAR

4.1 Introduction

It is desirable to verify the performance gains obtainable with SLR processing using sea data. However, multibeam sea data has not previously been conveniently and economically available. The Applied Research Laboratory of The University of Texas (ARL) has not implemented a digital sonar that processes stave recordings through a digital beamformer and a signal processor. To operat within funding constraints the SLR processor was implemented on ARL's CDC 3200 digital computer. This implementation was also a useful test of the adaptability of the SLR processor to, a smaller, less sophisticated machine.

4.2 Implementation of SLR Processor

ARL's digital sonar furnishes one channel of amplitude samples that are indexed by range and bearing. Since there is no Doppler information or separate high-Doppler channel, the techniques developed during the present contract were not used in this SLR processor. The logical structure of the low-Doppler SLR processor was used without change. The program receives incoming data samples, compares them with previously stored tracks to form possible tracks, calculates joint log likelihood ratios, and makes statistically significant decisions.

The major required modifications concerned the technical implementation of the logical structure into a computer code. In order to understand why these changes were necessary, it is instructive to review some of relevant facts about TRACOR's UNIVAC 1108 computer and ARL's CDC 3200 computer. The 1108 is a large storage machine with a sophisticated compiler, while the

3200 is a medium storage machine with a good, but less efficient compiler. The 1108 is faster in execution of programs.

In development of the SLR processor the flexibility and large storage capacity of the UNIVAC 1108 has been used extensively. The TRACOR program consists of a number of subroutines, each accomplishing specialized tasks. While this sacrifices some program efficiency, it allows flexible programing and the easy incorporation of newly developed procedures. For the 3200 implementation, greater efficiency and compactness was achieved by rewriting the SLR processor. The revised version consists of one large program of in-line code and three short subroutines that contain code that is repeated in several places in the main program. Where possible, flexibility to expand to more dimensions was incorporated; however, this expansion is not as easy as with the TRACOR program.

Another machine difference dictated reformatting the log likelihood ratio associated with each track as well as several processor parameters. The 3200 requires two core storage words to accommodate a decimal representation of a real number, while the 1108 requires only one. Since a joint log likelihood ratio is a decimal number and must be stored for each track held in core storage, doubling the required storage would increase storage requirements considerably. Integer numbers require only one word of storage in the 3200; therefore, the joint log likelihood ratios are multiplied by a scale factor and translated to an integer prior to storage. Most decimal parameters are treated in a similar manner.

A description of the program and its required inputs is given in Appendix B as well as a complete listing and flow chart.

4.3 Description of ARL's Data Processing

ARL's digital sonar consists of recorded stave outputs, a digital beamformer, an envelope detector and an integrator for each fixed beam. In addition, a general purpose digital computer further processes the data by calculating the mean, standard deviation and median of the samples contained in an annulus centered around each sample in the sonar output array. If the sample divided by the mean of the data in the annulus (this is, a normalized sample) is less than a specified threshold, for example 1, or if the sample is less than any of the samples contained in adjacent range and bearing location it is rejected. The end result of the above processing is a reduced number of samples that are local peaks and which exceed a certain threshold.

TRACOR's mathematical model of ARL's system output assumes Gaussian statistics out of the beamformer. This implies that the envelope detector yields Rayleigh-Rice statistics for signal-plus-noise samples and Rayleigh statistics for noise-only samples. Using this result, the statistics at the output of the integrator may be found by an appropriate number of convolutions of the above density function. To account for the normalization process, i.e., dividing one random variable by another, the density function of the ratio of the data sample to the mean of the annulus was found by numerical integration. Although the above process closely models the physical processor, it was found that the computed results do not fit the observed data for any bandwidth-averaging time ($\beta\tau$) product between 1 and 13 (which is the range of variation of $\beta\tau$ as the data varies from reverberation to noise limited condition). That is, if the data were from a ping cycle in which a CW signal had been transmitted, then one would expect the $\beta\tau$ product to range from 1 (reverberation limited data) to 13,

(noise limited data). Since linearly frequency modulated pulses were used to generate this data, the $\beta\tau$ product should not vary unless the passband of the receiver does not match the frequency band of the signal. The observed deviation may be caused by several factors, such as variation of envelope statistics caused by the approximate envelope detector used in the ARL digital sonar, variation in the assumed Gaussian input statistics due to at-sea conditions, round-off errors in the numerical calculations involved in the model, passband variations in the receiver, and data quantization in ARL's digital sonar (the above model assumes no quantization of input data). Nevertheless, by averaging the probability distributions for $\beta\tau$ products of 1 through 13 a close approximation to the observed distribution was obtained, as is shown in Fig. 4-1.

4.4 Derivation of Log Likelihood Ratio

4.4.1 Development of the Model for Output Statistics -

The model is based on a step by step transformation of the input statistics. It is assumed that the output of beamformer is Gaussianly distributed when noise alone is present and is Gaussian plus a sine wave for signal plus noise. The bandwidth of the data is approximately 400 Hz. It is well known that the envelope of the assumed wave form is distributed as a Rayleigh-Rice random variable,

$$f(x) = \frac{x}{\psi_o} \exp\left[-\frac{x^2 + a^2}{2\psi_o}\right] I_o\left(\frac{xa}{\psi_o}\right) \qquad (4\text{-}1)$$

where

ψ_o = noise power prior to envelope detection

a = the peak amplitude of the signal.

For mathematical simplicity it is assumed that ψ_o is identically 1. If a is zero then noise alone is present and Eq. (4-1) reduces to the familiar Rayleigh density.

TRACOR

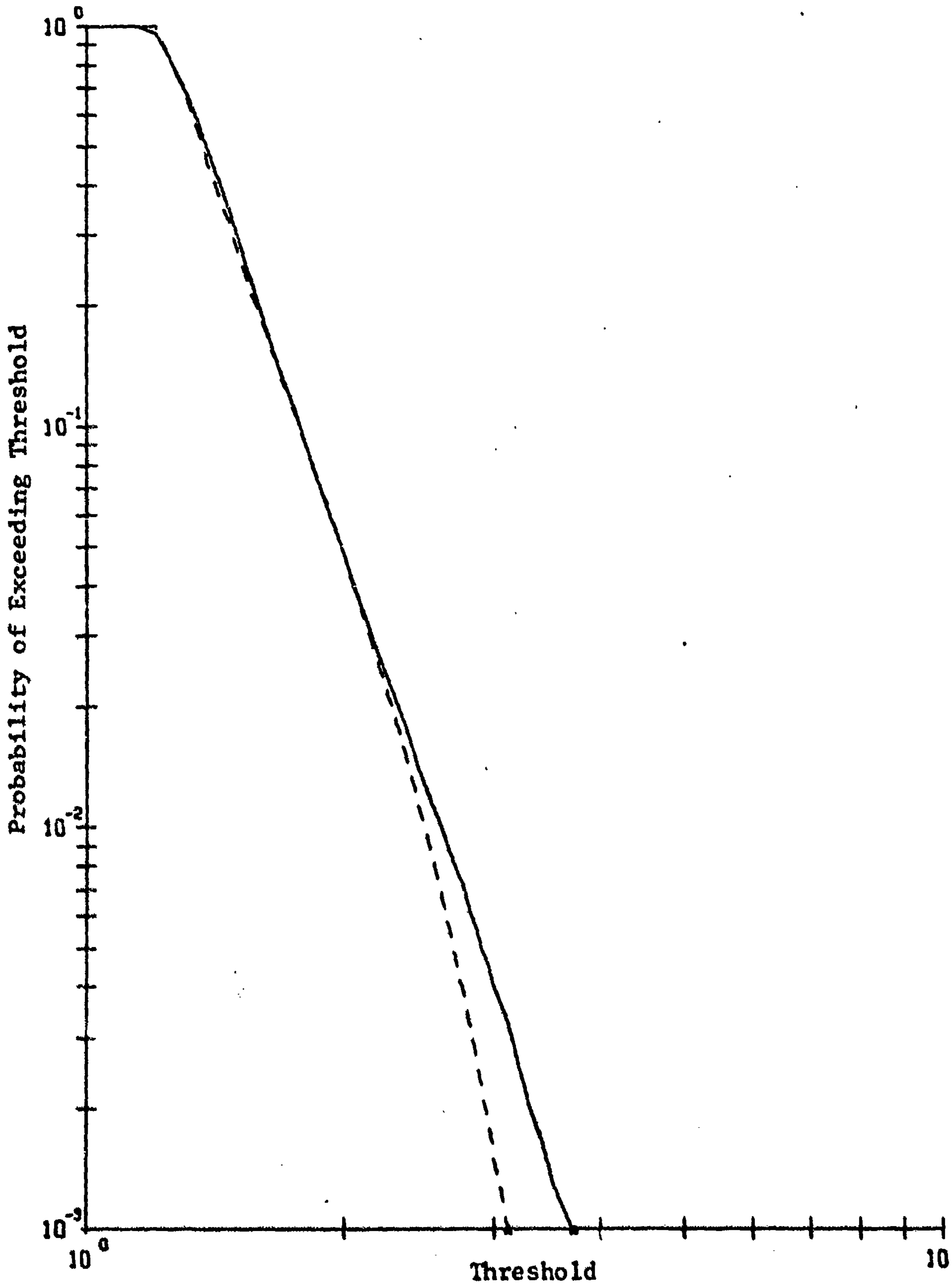


FIG. 4-1 - COMPARISON OF THE PROBABILITY OF EXCEEDING THRESHOLD VS THRESHOLD FOR ARL'S OBSERVED DATA (SOLID) AND TRACOR CALCULATED RESULTS (DASHED)

$$f(x) = x \exp(-x^2/2) \tag{4-2}$$

Equations (4-1) and (4-2) give the statistics of signal-plus-noise after envelope detection.

The signal is 30 ms long and the digital sonar integrates for 32 ms. If at any point in time the $\beta\tau$ product is equal to N, then N independent envelope samples are being summed. The density of these samples may be found by convolving the signal-plus-noise or noise-alone density N times. The complicated structure of Eqs. (4-1) and (4-2) preclude carrying out the convolutions in closed form if N is even moderately large. The convolutions were accomplished by making a discrete approximation to Eqs. (4-1) and (4-2) and then taking the inverse discrete Fourier transform with an FFT algorithm to form an approximation to their characteristic function. It was then possible to raise the characteristic functions to the power N yielding the characteristic function of the N-fold convolution of Eqs. (4-1) and (4-2). The required density function was determined by taking the discrete Fourier transform of the calculated characteristic function. The RMS error of the above process is about 1.5×10^{-7}*. This is good over most of the range of the density function but leaves much to be desired in the tail of the distribution where probabilities are of the same order of magnitude as the RMS error. To overcome the problem, the Edgeworth expansion** of the sum N variates distributed

---

*James F. Ferrie, et al, "Comparison of Four Fast Fourier Transform Algorithms," Naval Underwater Systems Center, Newport, Rhode Island, 3 June 1971.

**M. G. Kendall and A. Stuart, "The Advanced Theory of Statistics," Vol. I, Hafner, New York (1963) pp. 157-160.

according to Eqs. (4-1) and (4-2) was found. This asymptopic expansion was found to be quite good for very small values of N and was used in the later numerical integration routines. The Edgeworth expansion is basically a normal approximation to the distribution of the sum, but contains correction terms based on the higher order moments of the original density function. As the number of random variables summed increases, the correction terms become quite small. While it was necessary to use the Edgeworth expansion of an individual averaged output sample, the Gaussian approximation for the average of the samples in annulus was quite good, since 64·N independent Rayleigh-Rice samples had been added together. Hence, for any $\beta\tau$ product the density functions for the output sample and/or the mean of the annulus around it were known.

With the above information the density of the ratio of the sample value, x, to the mean of the samples in the annulus, y, may be calculated by considering

$$
\begin{aligned}
P(t \le T) &= P_r \left(\frac{x}{y} < T\right) = Pr\,(x < Ty) \\
&= \int_0^{\infty}\int_0^{Ty} f(x)\, g(y)\, dx\, dy \\
&= \int_0^{\infty} F(Ty)\, g(y)\, dy
\end{aligned}
\tag{4-3}
$$

where

g(y) is the density function of the mean of annulus (assumed to be Gaussian for this study),

and f(x) is given by Eq. (4-1) or Eq. (4-2) for signal-plus-noise or noise alone, respectively. Regarding density, p(T), of the ratio $T = \frac{x}{y}$ is given by

$$p(T) = \frac{d}{dT}\left(\Pr\left(\frac{x}{y} < T\right)\right) = \int_0^\infty y f(Ty)\, g(y)\, dy. \qquad (4\text{-}4)$$

Equations (4-3) and (4-4) were calculated for a number of different values of T by numerical integration. The density of the ratio T was tabulated for a number of different signal-to-noise ratios and for $\beta\tau$ products from 1 to 13 in integer steps. Some problems were encountered in calculating the density function for the ratio T at higher values where the function values are very small and roundoff error became important.

The final step in the process is choosing only local peaks; that is, only the samples whose ratio exceeds those ratios associated with eight adjacent range and bearing resolution cells. If $f_{S+N}(x,a)$ is the density of the ratio for signal-plus-noise and $f_N(x)$ is the density of the ratio for noise alone, then the required probability, P_{PK}, is

$$P_{PK}(x > T) = \int_{T}^{\infty} f_{S+N}(x,a) \int_{-\infty}^{T} f_N(x_1) \int_{-\infty}^{T} f_N(x_2) \ldots \int_{-\infty}^{T} f_N(x_8)\, dx_8 \ldots dx_2\, dx_1\, dx$$

$$= \int_{T}^{\infty} f_{S+N}(x,a)\, F_N^{\,8}(x)\, dx \qquad (4\text{-}5)$$

where

$F_N(x)$ is noise cumulative distribution of the ratio

For signal-plus-noise Eq. (4-5) may be determined by numerical means. For the noise along case, however, Eq. (4-5) becomes

$$P_{PK}(x > T) = \int_{T}^{\infty} f_N(x)\, F_N^{\,8}(x)\, dx$$

$$= \frac{1}{9} F_N^{\,9}(T) \qquad (4\text{-}6)$$

The sonar data furnished TRACOR by ARL had been subjected to peak selection and was conditioned on the crossing of a threshold on the ratio of about 1.2. For the purpose of data comparison this complemented conditional probability distribution was calculated and is given by

$$PR(x > T \,/\, x > 1.2) = \frac{PR(x>T)}{PR(x>1.2)},\ T \geq 1.2.$$

Figure 4-2 gives the family of calculated complemented distribution functions for various $\beta\tau$ products. No one curve fits the observed data. However, the sorted data is taken from all parts of the ping cycle. The conditions are changing from reverberation limited conditions, $\beta\tau=1$, to noise limited conditions, $\beta\tau=13$. As an approximation to the ensemble average of the observed data the distribution and density functions for $\beta\tau$ products of 1 through 13 were averaged together. The results were shown in Fig. 4-1. The agreement is very good except at the lower probabilities where the previously mentioned inaccuracies in the numerical integrations and approximations become important. Since agreement is close and the mathematical model is based on known signal processing results, this model appears to be very adequate for noise alone. Unfortunately, very little signal-plus-noise data is available for analysis. Therefore, it is not possible to validate the signal-plus-noise part of the model. It should be pointed out that it depends on the same reasonable assumptions and that the SLR processor is reasonably insensitive to variations in assumed statistics.*
Hence, the signal-plus-noise part of the model should be good enough for the purpose of generating a log likelihood ratio.

4.4.2 Likelihood Ratio Equation for ARL's Output - The log likelihood ratio is defined as

$$L(x) = \log \left(p_{S+N}(x) \;/\; p_N(x) \right).$$

Using the derivatives of Eqs. (4-5) and (4-6) at three design signal-to-noise ratios of 6, 9, and 12 dB, the log likelihood ratio curves were plotted for the average of the various $\beta\tau$ products. The behavior of the curves near 1 or 4 is suspect

*Reeder, H. A., "Final Report, Simultaneous Likelihood Ratio Processing for Two Active Receivers," Vol. I, TRACOR Document T71-AU-9594-U, 25 August 1971.

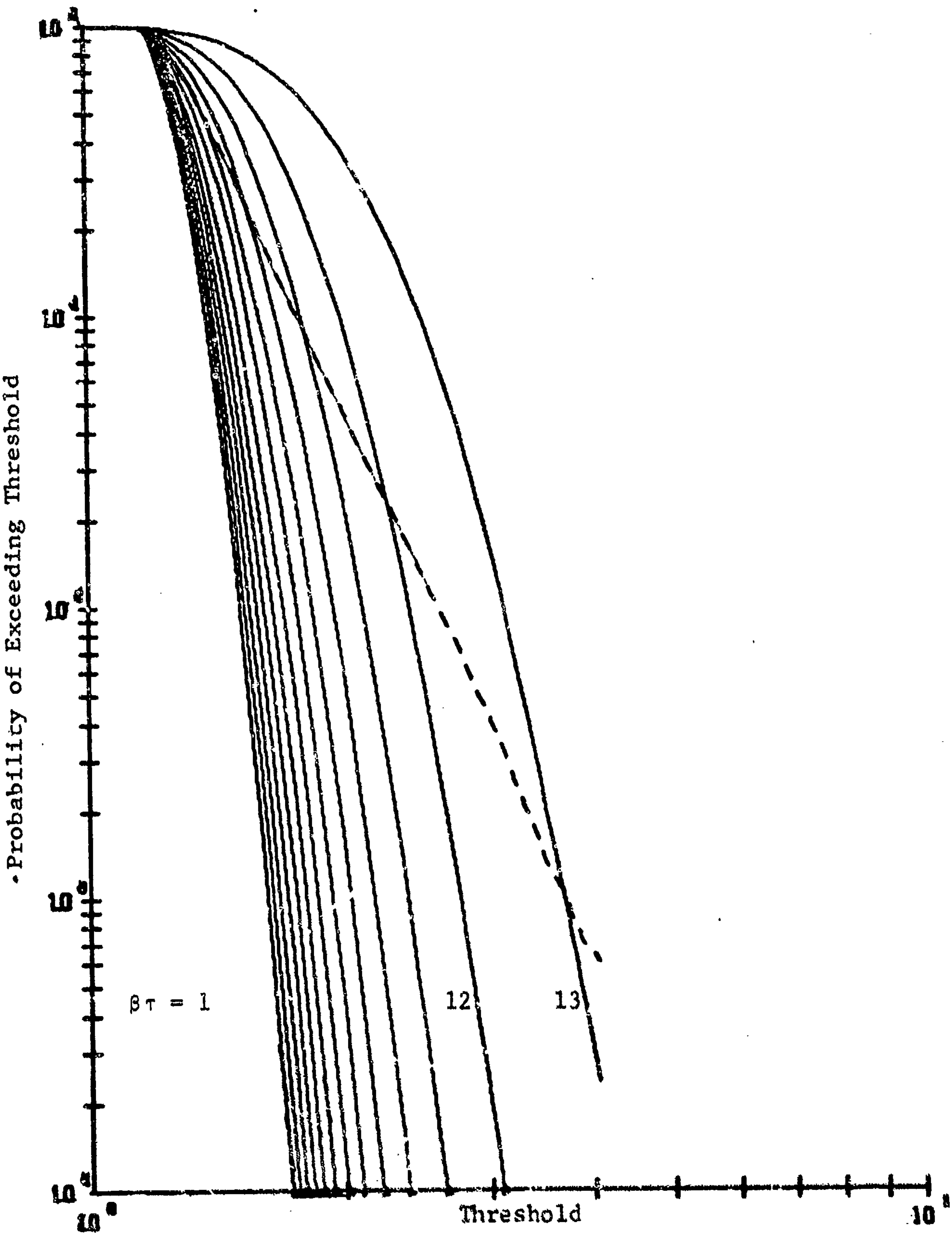


FIG. 4-2 - PROBABILITY OF EXCEEDING THRESHOLD VS THRESHOLD FOR β- PRODUCTS OF 1 THROUGH 13 (SOLID LINES) AND ARL'S OBSERVED DATA (DASHED)

because of computer round-off errors; however, the central parts of the curves are believed to be accurate. The results, along with a straight line least squares fit over the best part of the curve, are plotted in Figs. 4-3, 4-4, and 4-5. The straight line fits are summarized in Table 4-I.

TABLE 4-I

STRAIGHT LINE LEAST SQUARE FITS TO AVERAGE LOG LIKELIHOOD RATIO EQUATION

Design S/N	Slope	Intercept
6 dB	6.66	-9.23
9 dB	9.34	-15.40
12 dB	7.56	-15.32

The 9 dB design S/N seems to be the most reasonable starting place for SLR processing.

The model as derived appears to be reasonable; however, its accuracy may be improved by the expenditure of more time and money. For instance, by including more terms in the Edgeworth expansion more accuracy may be produced in the approximation. By extending the numerical integration region and subdividing it more finely greater accuracy may be obtained (it may be necessary to use double precision to prevent greater round-off error). These things would require greater computer expense as well as some reprogramming. More basic is the assumption of Rayleigh-Rice statistics rather than a fluctuating target model. There were neither time nor available funds to pursue these interesting questions. However, it is felt the present results are entirely adequate for the determination of an approximation to the log likelihood ratio equation for the SLR processor for the ARL's digital sonar output.

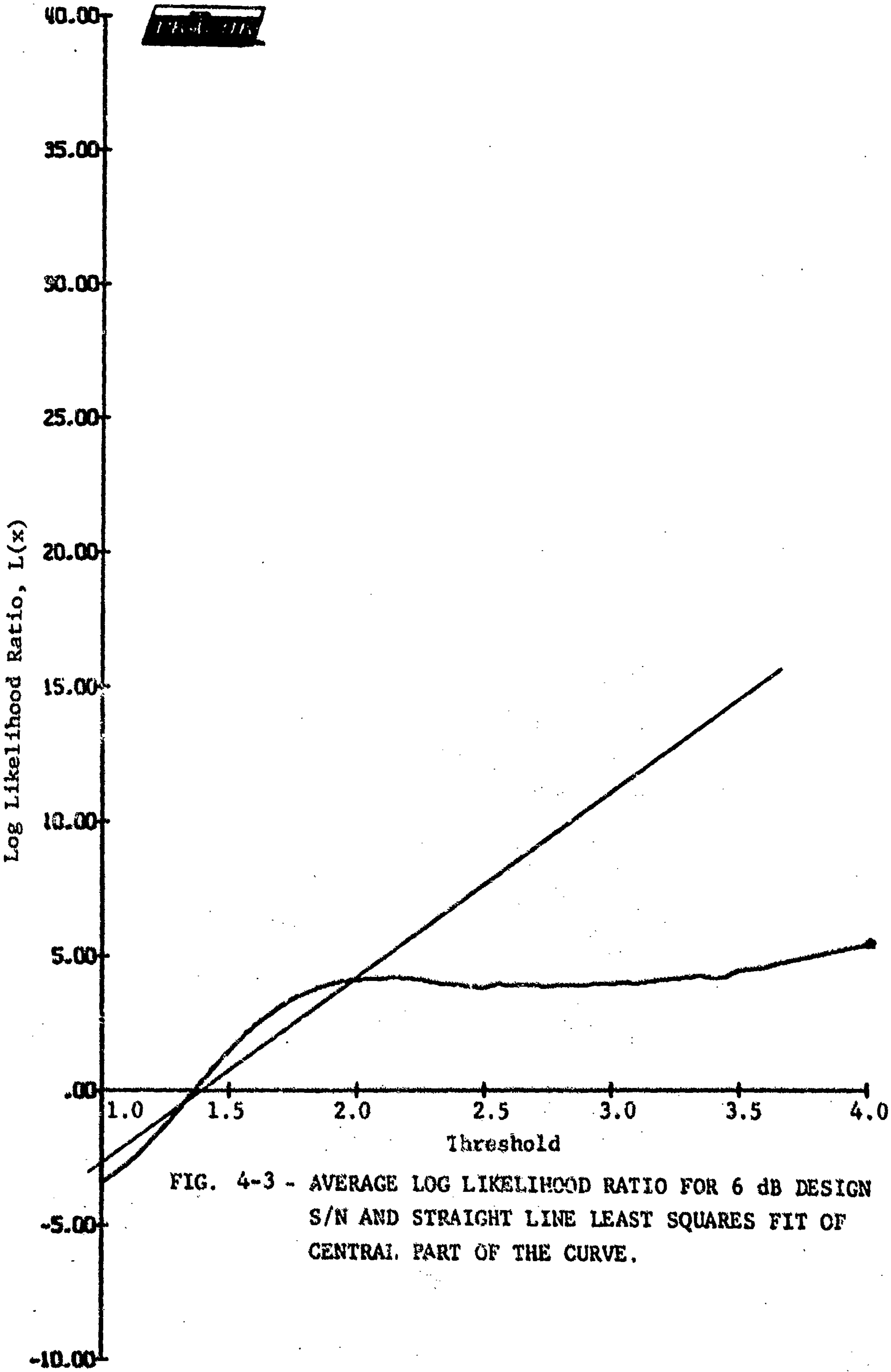


FIG. 4-3 - AVERAGE LOG LIKELIHOOD RATIO FOR 6 dB DESIGN S/N AND STRAIGHT LINE LEAST SQUARES FIT OF CENTRAL PART OF THE CURVE.

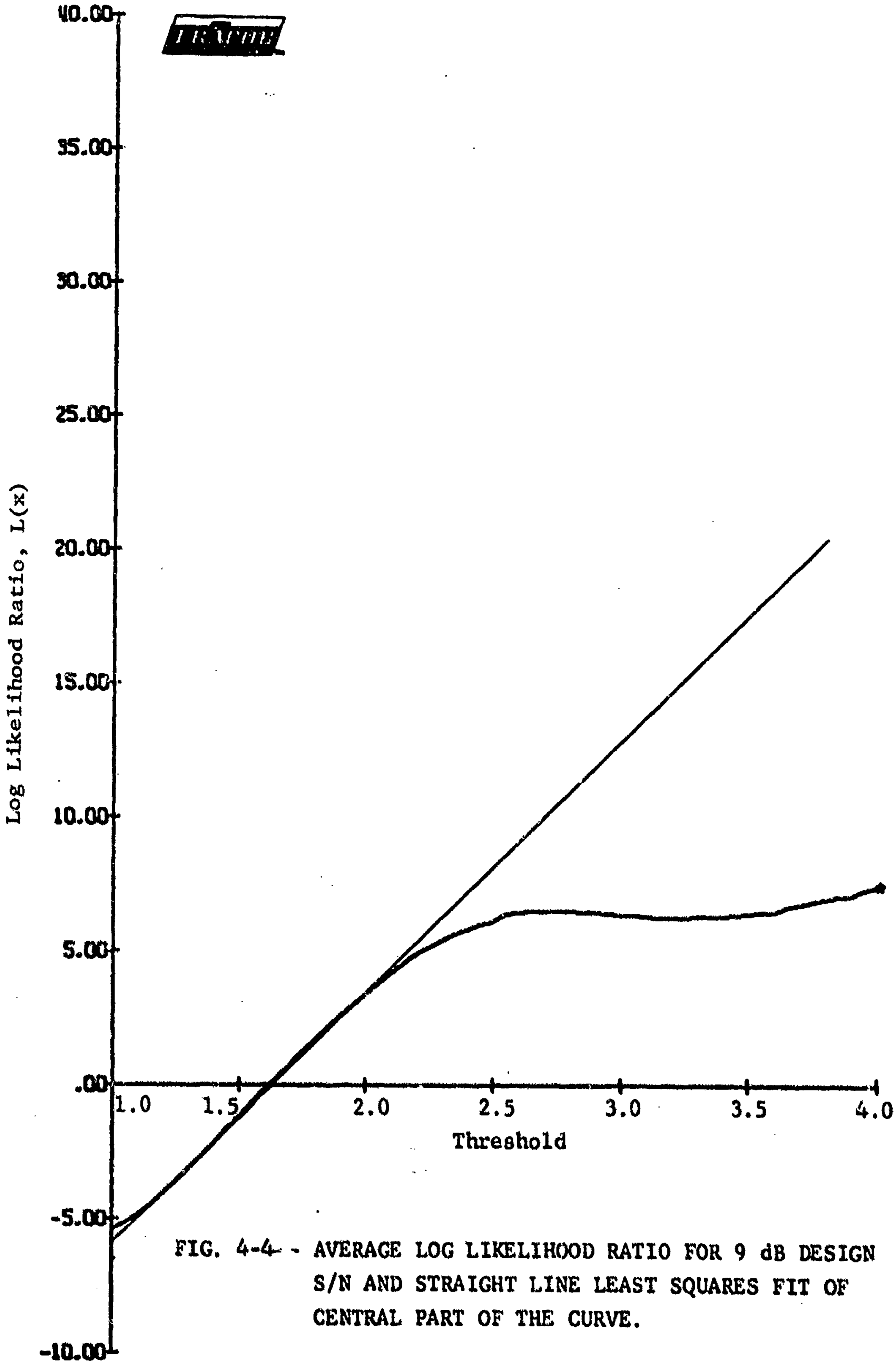


FIG. 4-4 - AVERAGE LOG LIKELIHOOD RATIO FOR 9 dB DESIGN S/N AND STRAIGHT LINE LEAST SQUARES FIT OF CENTRAL PART OF THE CURVE.

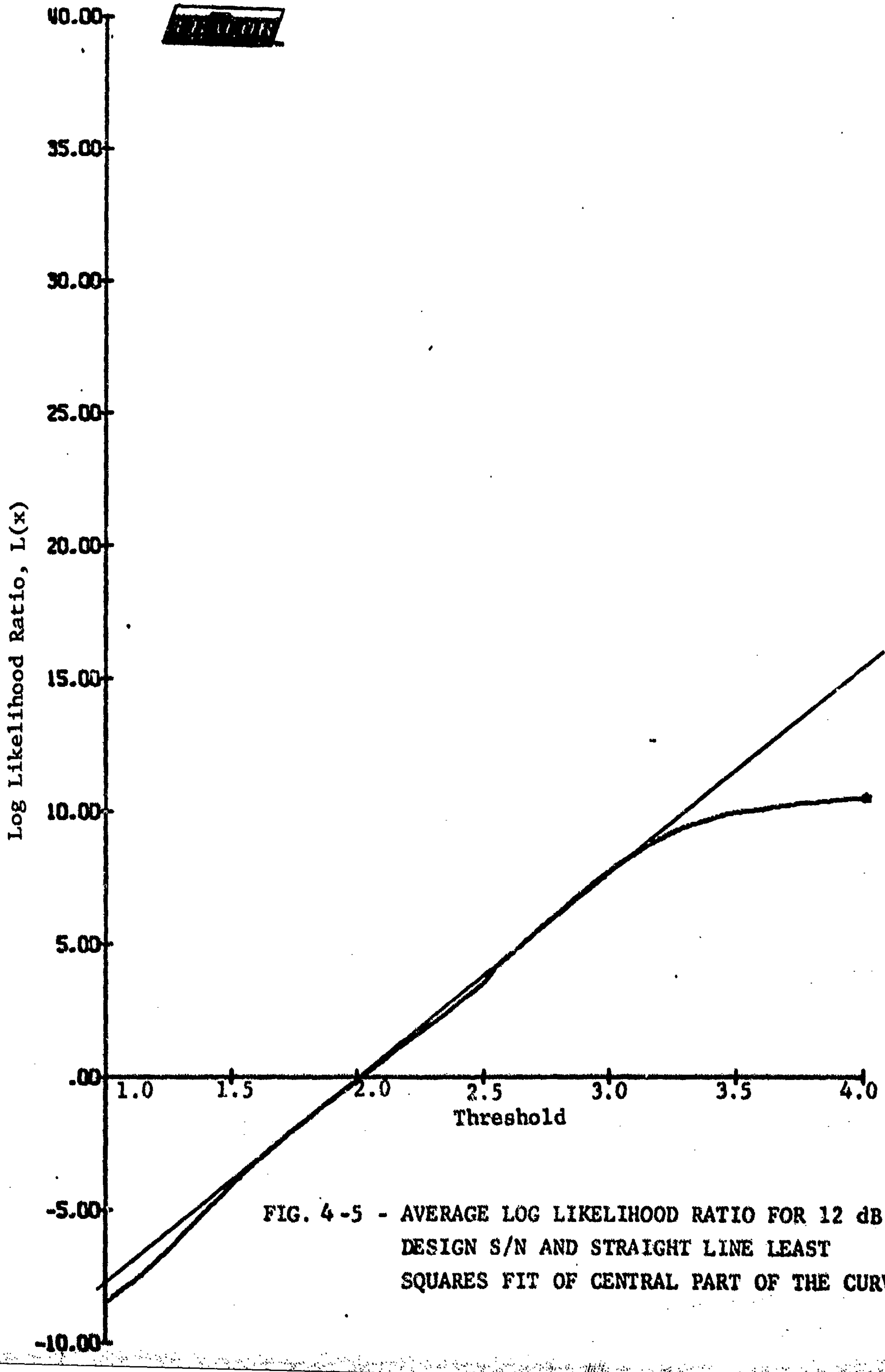


FIG. 4-5 - AVERAGE LOG LIKELIHOOD RATIO FOR 12 dB DESIGN S/N AND STRAIGHT LINE LEAST SQUARES FIT OF CENTRAL PART OF THE CURVE.

4.5 Choice Design Signal-to-Noise Ratio

The choice of a design signal-to-noise ratio is important to the performance of the SLR processor. Too high a value will cause poor detection performance; too low a value will cause excessive computer loading. In the previous section the derivation of the log likelihood ratio equations for three design S/N ratios were reported. These three equations were used to process the same data run (ARL run number 4). The outputs of these runs were analyzed as well as the non-SLR data, that is, data unchanged from the ARL digital sonar. In each case the output data was thresholded at four levels to achieve specified probabilities of false alarm. The probabilities chosen correspond to a typical modern sonar and were used in a previous SLR study.* The measurements on the data include the ping number at which the target's output first crossed the specified threshold giving a measure of time to detect; the number of consecutive pings above the threshold after the first crossing, which indicates whether a detection by an operator is likely after the first crossing; and, finally, the total number of times the target output exceeded the threshold out of 50 ping cycles. The tabulated data, Table 4-II, indicate that a design S/N ratio of 9 dB is desirable from the standpoint of total crossings and first crossings. The computer loading for 9 dB is about 100 status units per ping cycle, which appears to be an acceptable number from a computer loading standpoint. For 6 dB the computer loading is higher, 220 status units, but has not reached unreasonable levels; whereas, 12 dB requires very little computer storage, 25 status units, other than program storage. For further studies a design signal-to-noise ratio of 9 dB will be adopted.

---

*H. A. Reeder and H. D. Record, "Computer Aided Detection (U)," TRACOR Document 68-1238-C, 19 November 1968, CONFIDENTIAL.

TABLE 4-II

EFFECT OF DESIGN S/N RATIO CHANGE

Design S/N	First Crossing Ping Number	Consecutive Pings Above Threshold After First Crossing	Total Number of Pings Above Threshold
$P_{FA} = 1.68 \times 10^{-3}$			
6	15	10	33
9	14	11	36
12	14	7	36
NON-SLR	14	4	31
$P_{FA} = 1.8 \times 10^{-4}$			
6	16	5	26
9	15	6	30
12	15	4	24
NON-SLR	15	1	17
$P_{FA} = 1.4 \times 10^{-5}$			
6	17	1	17
9	17	2	21
12	31	2	17
NON-SLR	40	2	4
$P_{FA} = 8.2 \times 10^{-7}$			
6	36	2	11
9	35	2	12
12	36	2	14
NON-SLR	47	1	1

The data in Table 4-II may also be used to compare the performance of the SLR and non-SLR. For the lowest display level (highest false alarm probability, $P_{FA} = 1.68 \times 10^{-3}$) the time (in number of pings) to first crossing of the threshold is about the same but the persistence of marking after the first crossing is significantly better for the SLR. On the other hand, the total number of crossings, while greater for the SLR, is not significantly greater. For the next level ($P_{FA} = 1.8 \times 10^{-4}$) the persistence is again significantly greater for the SLR both after the initial crossing and in addition for the entire run. For the two highest (two lower false alarm probabilities, $P_{FA} = 1.4 \times 10^{-5}$ and 8.2×10^{-7}) the persistence after the first crossing is short for both SLR and non-SLR but the time of the first crossing is much sooner for the SLR than the non-SLR. Again, the total number of threshold exceedings is much higher for the SLR than the non-SLR.

The data in Table 4-II also indicate that the SLR being thresholded at $P_{FA} = 1.8 \times 10^{-4}$ performs very similarly to non-SLR at $P_{FA} = 1.68 \times 10^{-3}$; similar performance also exists with the SLR set at $P_{FA} = 1.4 \times 10^{-5}$ and the non-SLR set at $P_{FA} = 1.8 \times 10^{-4}$. This means that the SLR can allow the operator to reduce the false alarm probability (and hence, rate) significantly (approximately an order of magnitude in this case) while maintaining the same detection performance. The operator then has an easier job in evaluating the reduced amount of data, since there is significantly less of it for him to process.

4.6 Noise Spoke Suppression

The principal problem has been processing the ARL digital sonar data. On some runs the noise background contains persistent noise spokes. The original ARL normalizer used, that is, dividing an output sample by the mean of the samples in an

annulus around it, did not remove these spokes entirely. The SLR processor tends to integrate the residual noise spokes in the same manner as signal-plus-noise, the result being a preponderance of spurious noise tracks. Therefore, the initial results of the SLR processing were disappointing for those runs in which noise spokes were pronounced.

ARL has subsequently implemented a noise spoke suppressor which has been used with some success. The first step of the suppressor is to average the sample to be normalized with the maximum adjacent sample, yielding an effectively longer averaging time. Then an estimate of the mean of the noise in that beam is subtracted from the overaveraged sample. This estimate is determined by finding the mean of those samples in the annulus that are also within one beam width of the sample to be normalized. (ARL reports that an amplification factor of two on this mean estimate yields better results for runs with pronounced noise spokes. However, for consistency between runs and to achieve an unbiased estimate (in the statistical sense) of the excess of the sample above noise level of the beam, an amplification factor of unity was used for all SLR runs.)

The implementation of this noise spoke suppressor improved the results of the runs where spokes were a problem. The improvement is relative. The overaveraging degraded the signal-plus-noise samples to some extent; but the overaveraging along with the subtracting the estimate of the noise mean in that beam reduced the noise spokes much more. Although the improvement was considerable, the problem of noise spokes was not completely eliminated.

5.0 SLR DISPLAY STUDY

5.1 Introduction and Purpose of Study

This section describes details and results of an observer study carried out by TRACOR personnel to validate experimentally a processor that utilizes as its basic tool the Sequential Likelihood Ratio (SLR) test. The SLR processor is applied to the sonar signal processing sequence at the input to the operator display, i.e., just after signal processing has been completed on the input waveform. The processor depends upon the statistics associated with the processed waveform, and requires that probability distributions be derived for the noise-only background and for the signal-plus-noise background. By applying the theory of maximum likelihood, the processor has the capability to perform simple hypothesis testing on each time sample it receives; that is, to test the hypothesis H_o: no target present vs H_1: target present. The processor can improve its performance, taking advantage of computer speed and storage capabilities, by integrating over several, or many, time samples and performing sequential likelihood ratio tests. This type of ping-to-ping integration is performed internally, with the decision to display the integrated data dictated by the build-up of the likelihood ratios. The operator is presented with only that data which has been accepted under hypothesis H_1, resulting in a less cluttered display than if the same integration had been performed visually. This factor becomes important when the operator is required to view a display containing many beams or channels of data.

Before implementing such a processor, one would like to evaluate its performance under operational conditions. This study attempts to do this by presenting to trained observers the outputs of both an SLR processor and a non-SLR processor that have operated on actual recorded sea data. The data were furnished

by the Applied Research Laboratory (ARL) of The University of Texas. ARL processed a number of stave recordings made at sea through a digital beamformer and signal processor, making available the outputs on digital tape. These tapes then provide a multibeam realistic data base for the processor evaluation.

Results from the observer trials can be used to validate the performance of the SLR processor as compared with theoretical evaluations already obtained.

5.2 Description of TRACOR Display Facility

5.2.1 Hardware - The TRACOR display facility utilized for the observer study consisted of a magnetic tape transport, a system control unit, a core memory, and eight black and white television monitors.

The general operation of the system is as follows:

1. Magnetic tapes containing the information to be displayed are generated off-line using the UNIVAC 1108 computer facility. For any particular display problem, the programmer must create a driver program that utilizes standard software routines developed especially for the display facility. This program creates the tapes which contain digital data representing the output of a sonar signal processing sequence.

2. The digital tape transport is then used to transfer information from the magnetic tape to the core memory.

3. The system control unit sequentially scans the core memory and transfers the information to the television display monitors. This is done at an equivalent 60 Hz rate so that a flicker free display is obtained.

5.2.2 Software - The software package, together with the driver program, provides extremely flexible control over the instrumentation. The software makes available 122,760 independent addressable locations or spots on the CRT display, a matrix with dimensions 330 x 372. Each spot may assume any of eight grey levels, from black to white. The software establishes the matrix as the first quadrant of a Cartesian coordinate system. Spots may be displayed by specifying the coordinates of the spot and its intensity. Subroutines have been developed that use certain of the spots to display lines, arc segments, alphanumeric characters, and bounded polygonal regions.

5.3 Display Format

The displays were generated in a B-scan type format retaining a five ping history. The vertical axis, considering the face of the display as the first quadrant of a Cartesian coordinate system, contained 372 grid points, of which 333 were used (allowing for margins at the top and bottom) to define the range sector. The displays presented the minimum range at the bottom of the grid, with the maximum range being shown at the top. No raster lines or tic marks were used on the vertical axis to label or define the actual ranges under consideration. The horizontal axis, containing 330 grid points, defined the beams, or bearing sectors. Twenty-four beams were displayed, each defining a sector 7.5 degrees in width. Raster lines were drawn to define the beams. The even numbered beams were labeled at the bottom margin; beam 24 was excluded due to space limitations. The ping histories were carried along the horizontal axis; thus, a given beam could contain up to five marks for any given range bin. The updating procedure worked from left to right, with the current ping information displayed at the left-most portion of the beam and the four past pings stored to the right. As a current ping was being displayed, the oldest ping (i.e., five pings old) was

eliminated, with the remaining four pings of data each shifted one position to the right. Additionally, the current ping number being displayed was recorded in the upper margin at the left hand side of the display.

Care was taken in the setting up of the display driver program to insure that the target track always appeared somewhere on the grid, but that it did not appear in the same place (for example, centered) for each run.

5.4 Data Processing Sequence

This section describes the data processing that was necessary to convert the recorded sea data into a form suitable for display.

ARL generated a series of tapes that contained the outputs of a digital beamformer and signal processor. Actual sea data was processed through this simulated AN/SQS-23 which has a multibeam capability, with the resulting outputs being range, bearing, and amplitude information for each of 48 beams on each ping cycle. Ideally one would like this data to contain information not only about the noise background but also about the signal or target background. However, since controlled recordings of describable target behavior are difficult to obtain, ARL provided simulated target recordings by injecting a target signal into the recorded noise data. This resulted in 12 fifty-ping runs with a target signal that was varied from approximately 0 dB signal-to-background ratio (at the processor input) at the start of the run to 18 dB at the end of the run with the S/N level increasing 2 dB every 5 pings. The target tracks associated with these 12 runs were of the straight line constant speed type. One additional run was generated in which the S/N was held constant over the entire ping sequence but the track was that of a maneuvering target.

TRACOR's CDC 3200 FORTRAN version of the SLR processor was used in the initial data processing of the ARL data tapes. This program, previously described in Appendix B, performs the necessary tape processing and bookkeeping functions, and accepts the data in a format standardized by ARL The program's two basic functions are the formation of log likelihood ratios and the application of a tracking algorithm. For SLR processing, each data sample that results in a likelihood ratio greater than a certain lower threshold is subjected to the tracking algorithm and possibly linked with other data samples in a manner dictated by the possible geometries of a target track. These linkages result in a formation of sequential likelihood ratios and are carried in the program (processor) until they ultimately fall below a lower threshold and are dropped out, or exceed an upper threshold and are displayed (or in our case written as output on a data tape). The important thing here is to control the number of linkages the processor must maintain in consideration of computer loading and time requirements. This control can be exercised by proper choice of the design signal-to-noise ratio used in calculating the likelihood ratios. After an initial examination of the data, 9 dB was determined to be a reasonable choice for this parameter and was used to process each run. The data output to tape by the processor consisted of the significant range-bearing-likelihood ratio samples at each ping cycle. Those data samples which were significant on a single ping basis were flagged so that they could be sorted exclusively when the non-SLR displays were generated.

The generation of the displays was done on the UNIVAC 1108 facility. A driver program was written utilizing special display software routines. The program transformed the data into a B-scan type format. Input to the driver program consisted of a processor output tape and control cards defining key

parameters. These parameters specified the run to process, the type of processor (SLR or non-SLR), the range sector, the beam numbers/bearing sectors, the thresholds used to control the brightness levels, and the background orientation.

All displays were generated with a 10 kyd range sector, the only differences from run to run being the starting or minimum range to be displayed. Due to the size of the displays, only 24 of the 48 beams of data could be shown at the same time. The program made it possible to select the 24 beams separately for each ping cycle and this feature was used to generate noise only displays by eliminating the beams containing the target echoes.

The program required seven threshold values to determine the brightness levels and clutter densities for each run. The thresholds were determined from an empirical examination of the background data, resulting in distributions for the probability of exceeding threshold vs threshold (likelihood ratio units) for each processor.

These distributions were initially generated for both the SLR and non-SLR processors by combining the data from all 13 runs. The thresholds obtained from this ensemble obviously will remain constant or fixed on a run to run basis. Additionally, it was decided to determine the distributions for each run separately, resulting in different i.e. adaptive, thresholding for each run. Sample plots of these distributions are shown in Figs. 5-1 and 5-2.

The final step in determining the proper display thresholds was to select a set of noise-only marking probabilities; that is, the probabilities that likelihood ratios obtained from the noise-only waveform will exceed certain levels. These probabilities are given in Table 5-I and are the same as those used in a previous

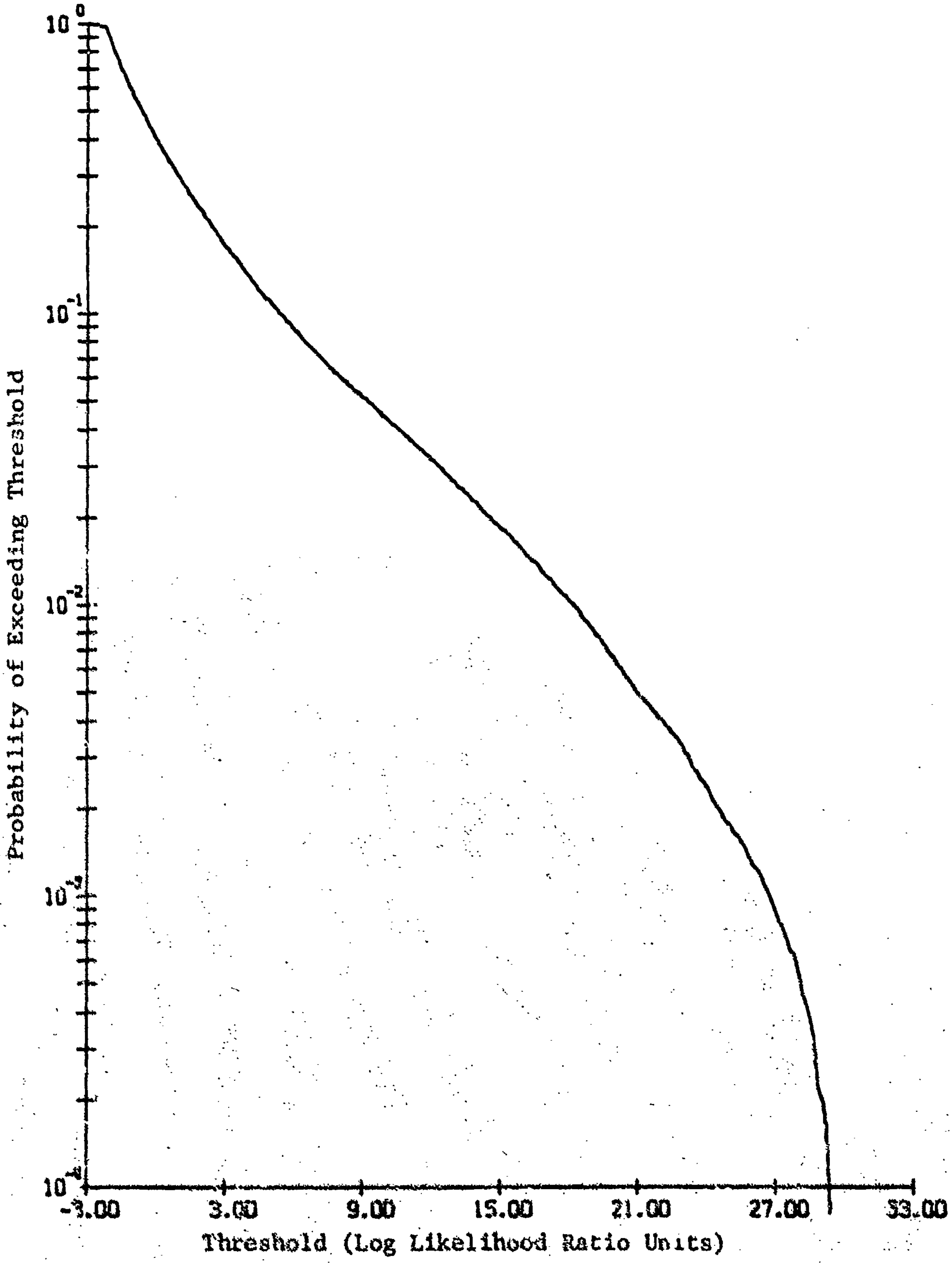


FIG. 5-1 - ENSEMBLE AVERAGE SLR PROCESSOR

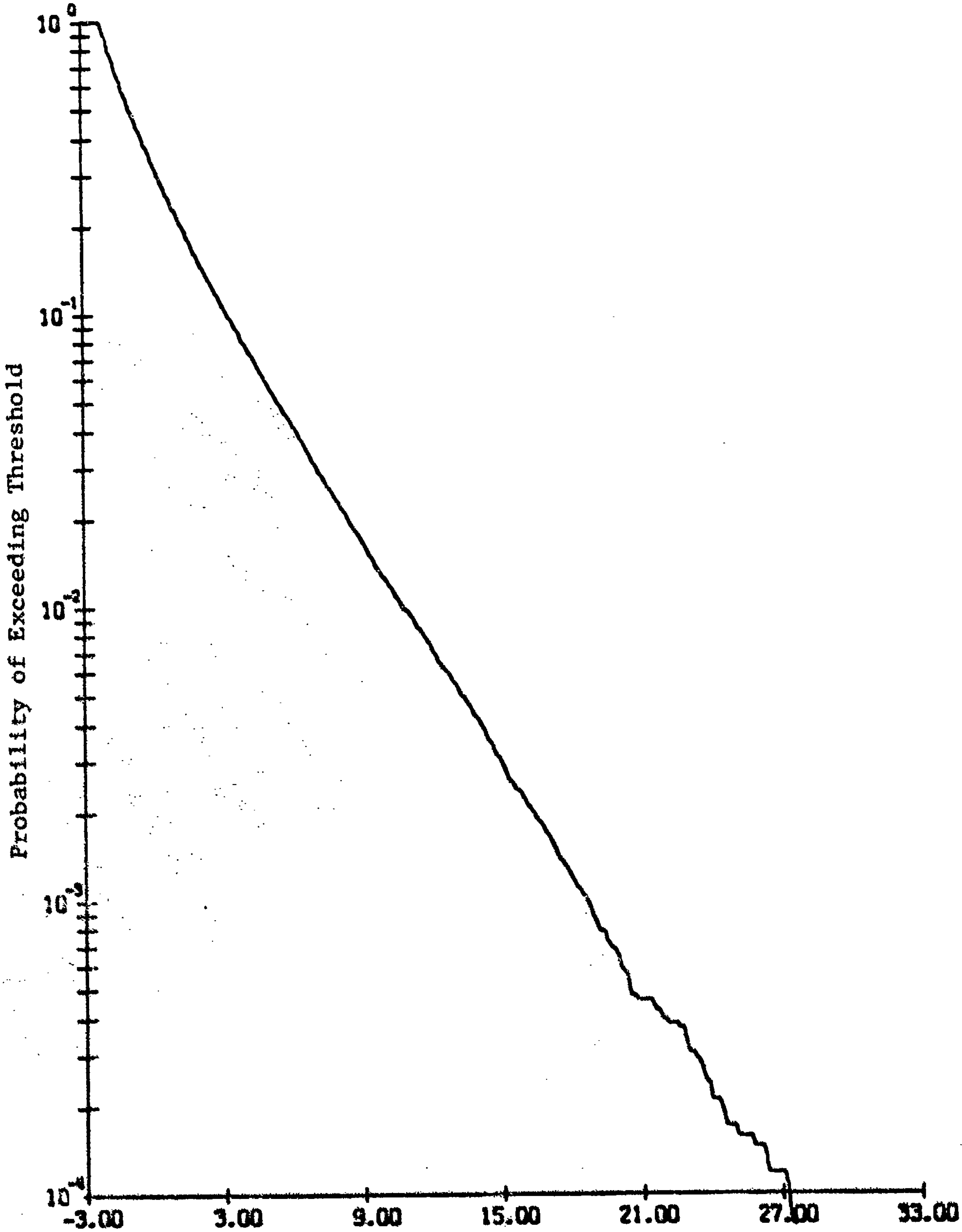


FIG. 5-2 - ENSEMBLE AVERAGE NON-SLR PROCESSOR

TABLE 5-I

PROBABILITIES OF NOISE MARKING
AT EACH INTENSITY LEVEL

Intensity Level	Probability of Marking With Intensity Level Ii or Higher
I_1	0.05
I_2	0.0102
I_3	0.00168
I_4	0.18×10^{-3}
I_5	0.14×10^{-4}
I_6	0.82×10^{-6}
I_7	0.33×10^{-7}

SLR display study*. It is important when comparing the SLR vs non-SLR processors that the clutter rates and brightness levels for the noise-only waveforms be the same. This can reasonably be accomplished by correctly setting the thresholds corresponding to the appropriate marking probabilities. Figure 5-3 illustrates the way in which the thresholds were obtained.

The plots shown represent the probability that the noise-alone waveform will exceed display threshold as a function of threshold at the output of the SLR processor and at the output of the non-SLR processor. By fixing the thresholds as shown in this figure, the displays will respond in the same way to both output waveforms, i.e., both displays will contain the same number of marks and the brightness distribution of the marks will be the same, if the probability scale is partitioned in the same way on both distributions. For example, the partition at Probability = 0.1 gives a threshold value for each distribution, T_1 for the SLR processor and T_1' for the non-SLR processor. No marks will be made if the waveform sample is less than T_1 in the SLR processor and less than T_1' in the non-SLR processor.

The next partition is shown for purposes of illustration at Probability = 0.01. This partition generates two more thresholds T_2 and T_2'. Similarly, T_3 and T_3' are determined at the third partition. The brightness distribution and the density of marking on the two displays will be the same if the waveform samples from the SLR processor lying between thresholds T_1 and T_2, between thresholds T_2 and T_3, etc. are marked at brightness level #1, brightness level #2, etc. while the waveform samples from the non-SLR processor are marked with the corresponding brightness

---

*H. A. Reeder and H. Record, "Summary Report on Computer-Aided Detection (U)," TRACOR Document 68-1238-C, Contract NObsr 93220, 19 November 1968, CONFIDENTIAL.

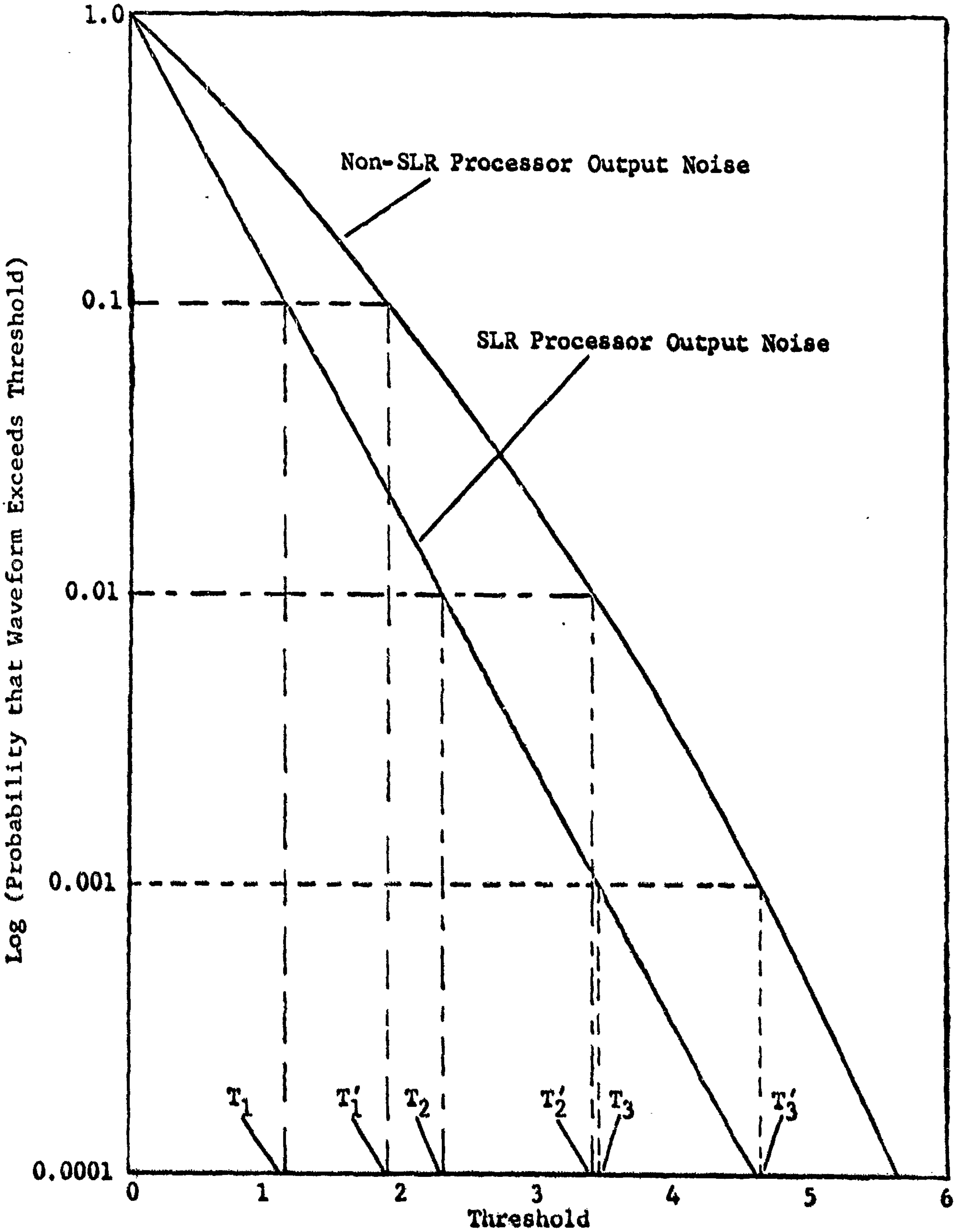


FIG. 5-3 - THRESHOLDS FOR EQUAL DISPLAY BRIGHTENING

if they fall between T_1' and T_2', between T_2' and T_3', etc. This description is for illustration of the method. In practice, seven brightness levels plus black were employed.

With the brightness structure and marking density identical in the non-SLR and the SLR displays, observer response to noise alone can be expected to be identical in an average sense as long as the observers operate under the same criterion in responding to the two displays. Variability from run to run will be expected because of the random nature of the detection process, but response will not be biased by differences in the noise marking on the displays of the non-SLR and the SLR processors.

The thresholds used in the study are presented in Tables 5-II and 5-III. One difficulty in setting the thresholds that control the upper brightness levels was that there were not enough data samples to provide empirical estimates corresponding to the smaller marking probabilities. In setting a level seven threshold corresponding to the brightest intensity, one needs at a minimum enough data samples to estimate $P_7 = 0.33 \times 10^{-7}$, or $1/0.33 \times 10^{-7} = 3 \times 10^7$ samples*. In fact, it was often necessary to estimate the last three intensity levels, especially for individual runs, due to a lack of data corresponding to those smaller probabilities. Unfortunately, this situation may penalize the SLR processor in that one of its characteristics is the ability to rapidly integrate log likelihood ratios in the presence of the target signal. <u>The need to discriminate strong target tracks at the appropriate intensities is necessary when comparing the two processors.</u>

---

*This means that somewhere around 360 fifty-ping runs are required, remembering that the normalizer uses a peak picking method that on the average admits about 1/9 of the original data to the processor.

TABLE 5-II

THRESHOLDS USED TO SET
INTENSITY LEVELS FOR THE SLR PROCESSOR

Intensity	Thresholds (Log Likelihood Ratio Units)						
	Run 1	2	3	4	5	6	7
1	1.12	0.72	-0.44	-2.94	-2.94	-0.70	-2.94
2	8.46	7.61	6.01	-0.49	-2.00	5.86	-1.80
3	18.26	18.00	16.40	3.50	-1.22	17.29	0.41
4	27.0	26.5	27.0	14.2	2.16	27.0	4.46
5	30.0	30.0	30.0	23.7	23.0	30.0	8.0
6	35.0	35.0	35.0	30.0	30.0	35.0	25.0
7	39.0	39.0	39.0	35.0	35.0	39.0	33.0

TABLE 5-II--Continued

Thresholds (Log Likelihood Ratio Units)

Intensity	Run 8	9	10	11	12	13	Ensemble Average
1	-2.09	-2.94	0.32	2.30	-2.94	0.49	-0.98
2	0.64	-0.69	7.32	10.36	-0.54	7.26	4.70
3	5.41	3.02	18.51	19.80	3.20	18.14	14.92
4	12.4	13.0	27.2	27.7	12.0	27.6	25.86
5	17.4	23.0	30.0	30.0	26.0	30.0	30.0
6	25.0	30.0	35.0	35.0	30.0	35.0	35.0
7	33.0	35.0	39.0	39.0	35.0	39.0	39.0

TABLE 5-III

THRESHOLDS USED TO SET INTENSITY
LEVELS FOR THE NON-SLR PROCESSOR

Intensity	Thresholds (Log Likelihood Ratio Units) Run 1	2	3	4	5	6	7
1	-0.28	-0.30	-1.10	-2.94	-2.94	-1.28	-2.94
2	4.41	3.97	3.57	-0.64	-1.84	2.90	-1.80
3	10.41	10.08	9.44	2.41	-1.24	9.51	0.41
4	17.2	16.1	19.1	10.8	1.18	21.5	4.46
5	28.0	20.0	28.0	20.0	20.0	28.0	8.0
6	33.0	25.0	33.0	25.0	25.0	33.0	15.0
7	37.0	30.0	37.0	30.0	30.0	37.0	20.0

TABLE 5-III--Continued

Intensity	Thresholds (Log Likelihood Ratio Units) Run 8	9	10	11	12	13	Ensemble Average
1	-2.22	-2.94	-0.44	0.49	-2.94	-0.39	-1.48
2	0.11	-0.84	3.71	5.00	-0.66	3.73	2.31
3	3.91	1.96	9.31	10.40	2.26	9.56	7.97
4	10.0	10.5	15.5	16.5	10.3	16.3	15.76
5	17.4	20.0	24.0	24.0	18.0	24.0	27.4
6	25.0	25.0	30.0	30.0	25.0	30.0	33.0
7	30.0	30.0	35.0	33.0	30.0	35.0	37.0

5.5 Design of the Observer Study

Thirteen separate runs were obtained from ARL and subsequently became the data base for the observer study. Each run was processed to generate four separate display runs, making fifty-two display runs to be shown. The design of the study can be considered to be a 2 x 2 factorial experiment arranged in randomized blocks. Describing it in this way, the treatments or factors consist of two processors, SLR and Non-SLR, and two thresholding methods, fixed and adaptive. In addition it was observed that the thirteen runs could be divided into two groups or blocks, determined by the background clutter densities; thus high density and low density runs. Actually this blocking was done to minimize possible learning effects that might occur from showing two or more displays of similar backgrounds in a particular sequence. An additional effort was made to disguise similar backgrounds. This resulted in creating half of the displays in reverse, i.e, as if the display itself were flipped over, giving the impression of different backgrounds and also resulting in different (reversed) target tracks. All of the displays with adaptive thresholding were treated in this manner. Although this reversal effect is therefore confounded with the thresholding treatment, it can reasonably be ignored in the analysis by assuming that the observers will operate in a "symmetrical" manner. The actual display sequence shown to the observers was generated in the following manner.

The runs numbered 1 through 13 were arranged in two groups corresponding to high and low clutter densities. There were seven high density runs and six low density runs. Since the numerical assignment of run numbers appears to have no significance, no randomization was performed on the ordered sequence 1-13. Instead treatment combinations of SLR-FIXED, SLR-ADAPTIVE, NON-SLR-FIXED, NON-SLR-ADAPTIVE were assigned the values 1-4, respectively. The

viewing sequence was determined by alternately selecting a high or low density run and obtaining a random number from one through four to determine the treatment combination. This process was repeated until all 52 displays had been assigned a sequence number.

After part of the viewing sequence had already been shown, it was decided to select three of the original 13 runs, remove the target tracks from them, and create new displays similar to the ones previously described. This was done in an attempt to better understand or measure the false alarm rates associated with the two processors under investigation. These 12 additional displays were interspersed with the original 52, after randomizing them as described previously.

5.5.1 Description of the Data Taking Process - A total of nine subjects participated as observers although several did not complete the entire sequence of runs. The subjects were trained as to the display format and the responses required of them on several 50 ping training tapes. These tapes contained a background of random noise over which a simulated target track was injected. This target track was characterized by a constant speed straight line geometry and a linearly increasing signal to background ratio throughout the 50 ping sequence. The subjects were told beforehand that there could be from one to four targets on any of the runs and were asked to respond to each ping cycle in the following manner: Record for each suspected target the nearest beam number, approximate range, and confidence level. The beams were defined on the displays by raster lines and beam numbers, but the subjects were required to estimate the range and place it roughly into one of three equal range increments, R1, R2, or R3. These were not defined on the display and so might vary from subject to subject on borderline cases. The confidence levels were defined by: 0 - no target

present; 1 - possible target present; 2 - probable target present; and 3 - positive target present. Although the subjects closely monitored each ping cycle it was only necessary for them to record any change in status that might have occurred from ping to ping, thus reducing significantly the amount of data taken. A typical response sheet taken from one subject on one run is shown in Fig. 5-4.

5.6 Presentation and Discussion of Results

Results from the study have been summarized from the tabulated responses and are presented in this section. Ensemble averages for runs 1-12 were generated for each processor, providing a significantly larger sample size than run to run comparison. Runs 1-12 are all similar in that the target track was simulated as a constant speed straight line nonmaneuvering course with a stepwise increasing signal-to-noise ratio. Results are also presented for run 13, which had a maneuvering target with a constant signal-to-noise ratio. The ensemble averages were derived from a tabulation of 405 observer responses, while run 13 had 30 responses. An additional breakdown shows the ensemble with 206 responses with the SLR processor and 199 with the non-SLR processor, while run 13 shows a breakdown of 18 and 12, respectively. It was decided to combine the results of the thresholding types after an examination of the data revealed no significant differences could be attributed to this factor*. As previously discussed, this result could be attributed to the inability to accurately estimate the upper threshold values.

Figure 5-5 presents the probability that the subjects correctly classified the true target track for a given run in the ensemble as a function of ping no. (i.e. time) for the SLR processor. The three curves presented represent the confidence rating

---

*See Table 5-X, page 80.

0- No Target Present
1- Possible Target Present
2- Probable Target Present
3- Positive Target Present

Date : 8-25-72
Tape Number : 5670
Run Number : 3
Subject Name: TAF

FIRST TARGET	SECOND TARGET	THIRD TARGET	FOURTH TARGET
5, 1, R2, B23	20, 1, R2, B5		
7, 2, R2, B23	21, 2, R2, B5		
8, 3, R2, B23	22, 3, R2, B5		
23, 0, R2, B23	24, 3, R2, B4		
	29, 3, R2, B3		
	34, 3, R3, B2		
	39, 3, R3, B1		
	49, 3, R3, B1		
False Target	Actual Target		

FIG. 5-4 - RESPONSE SHEET

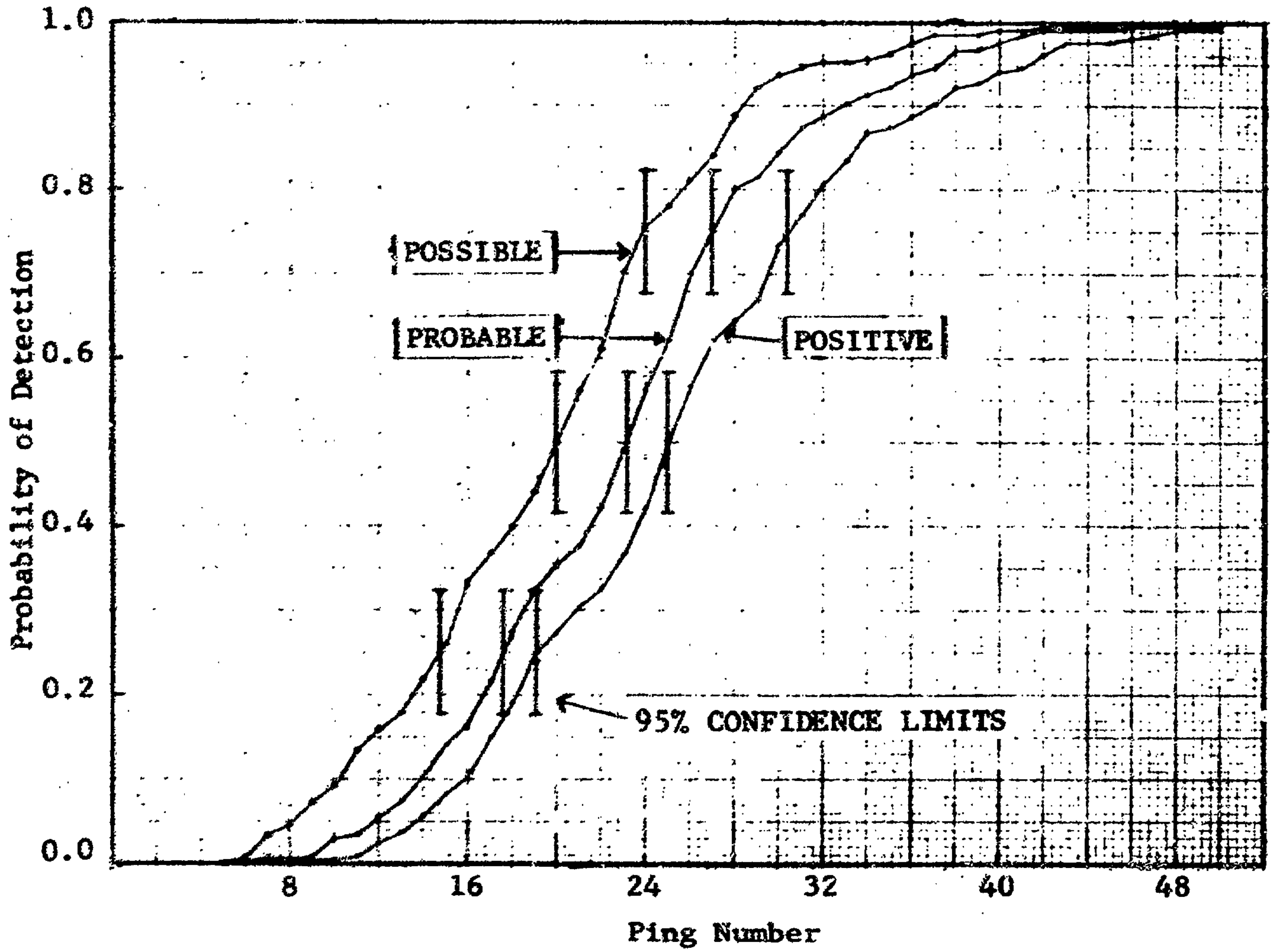


FIG. 5-5 - AVERAGE PROBABILITY OF DETECTION FOR SLR PROCESSOR AT THREE CONFIDENCE RATINGS, RUNS 1-12

associated with classifying the target track. A curve of a given confidence rating was generated by counting the number of correct calls of that rating or higher and dividing by the total number of possible correct calls, for each ping. 95% confidence limits were generated at selected probabilities. Figure 5-6 presents the same curves for the non-SLR processor, while Figs. 5-7 and 5-8 present these results for run 13 alone. Table 5-IV gives the earliest ping at which at least half the observers had correctly identified the target; i.e., 0.5 probability of detection, for a given confidence rating. This result is for runs 1-12 only.

Figures 5-9 and 5-10 summarize the incorrect or false target calls for the two processors averaged over runs 1-12. The top three curves give the average number of false target calls per ping by an operator for a given confidence rating. The bottom three curves present this same information in a cumulative format; that is, they give the average cumulative number of false alarms per ping by an operator for a given confidence rating. 95% confidence limits were generated at selected ping numbers for the cumulative false alarm curves. These curves were obtained from the displays that also contained the true target tracks. It is difficult to measure or estimate the false alarm rate from this data for several reasons. First, since the true target track becomes more easily detectable as the run progresses due to the increasing S/N ratio, the operators tend to stop calling any other potential target tracks. This is evidenced by the declining average number of false target calls after about ping 24 or so. Another problem with estimating the false alarm rate is that the displays present a five ping history of data to the observers, and it is difficult to provide a mathematical formulation of what would constitute a target call. For example, how many spots out of five must be marked and at what intensity levels before an operator will make a call. Realizing these shortcomings, a false alarm rate can be estimated from this set of data by obtaining the

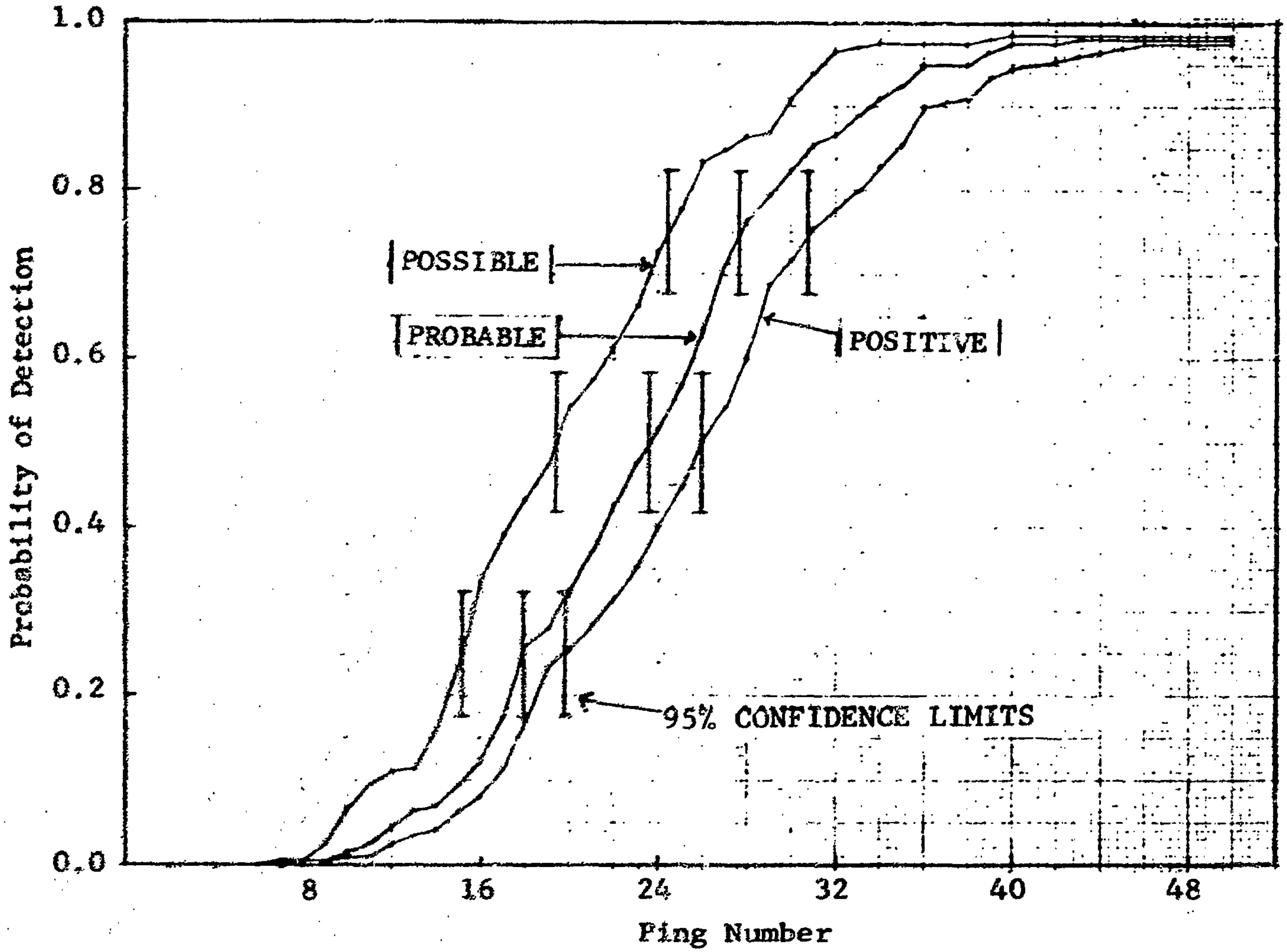


FIG. 5-6 - AVERAGE PROBABILITY OF DETECTION FOR NON-SLR PROCESSOR AT THREE CONFIDENCE RATINGS, RUNS 1-12

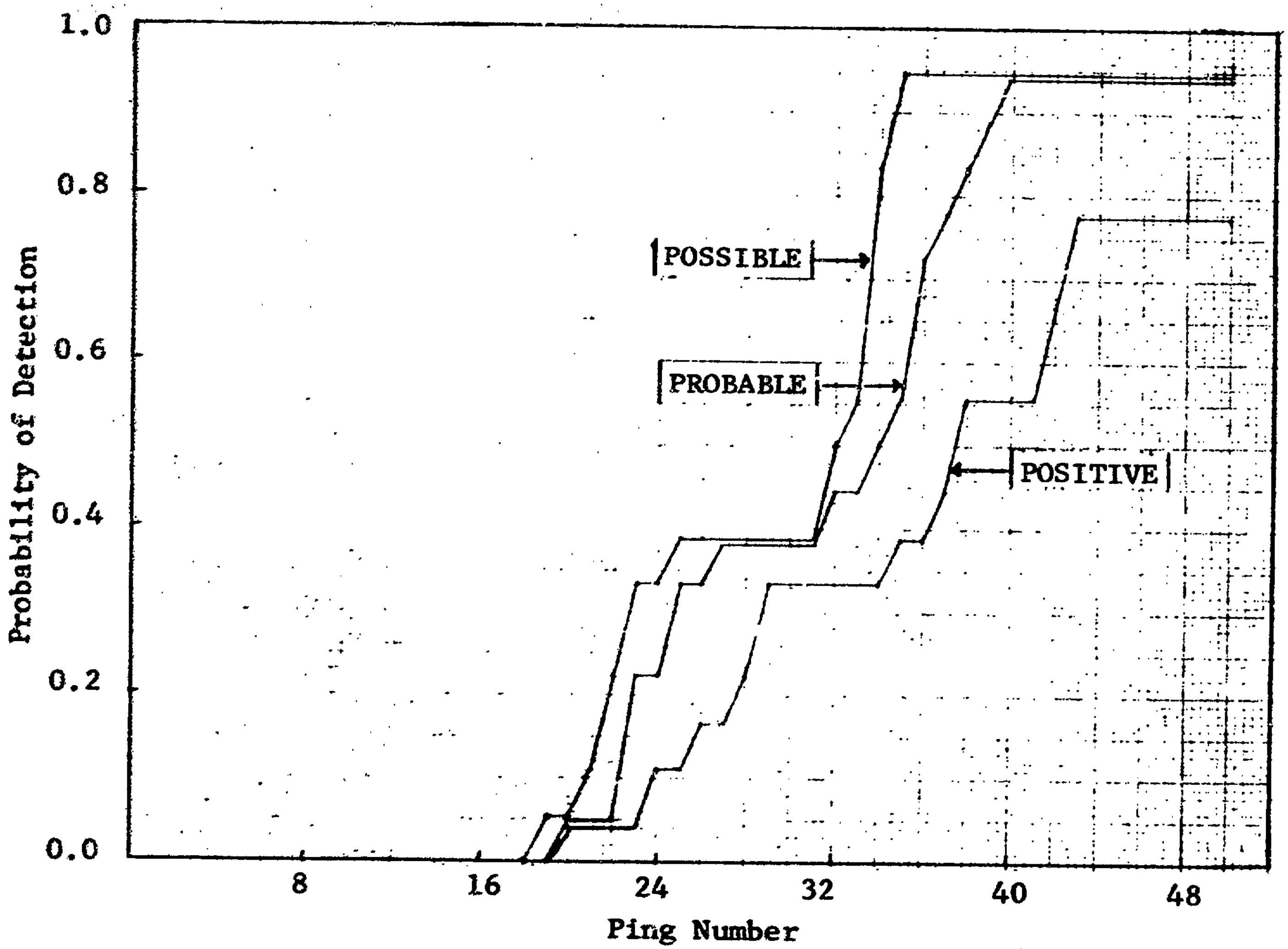


FIG. 5-7 - AVERAGE PROBABILITY OF DETECTION FOR SLR PROCESSOR AT THREE CONFIDENCE RATINGS, RUN 13

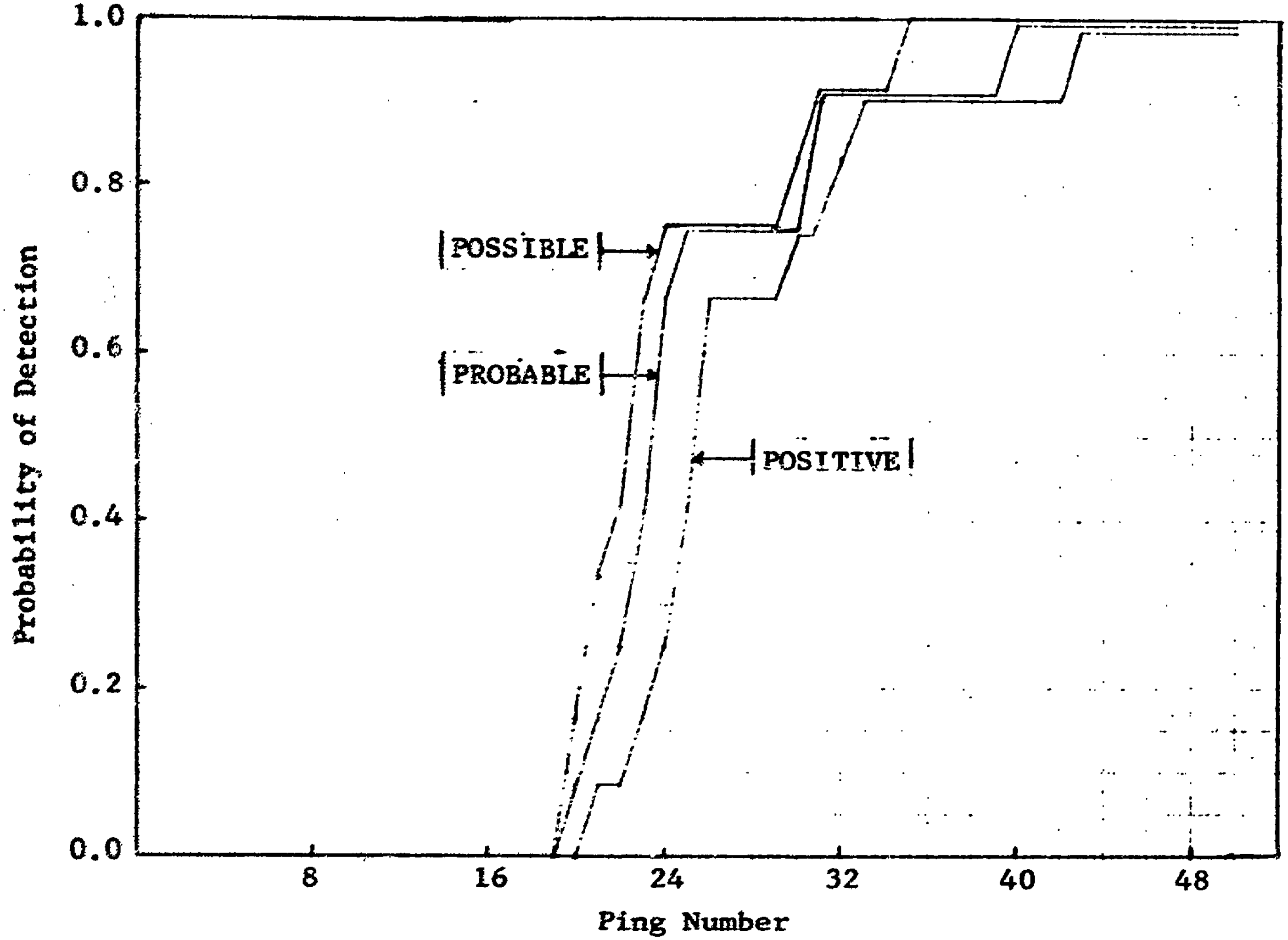


FIG. 5-8 - AVERAGE PROBABILITY OF DETECTION FOR NON-SLR PROCESSOR AT THREE CONFIDENCE RATINGS, RUN 13

TABLE 5-IV

ESTIMATED PING NUMBER AT WHICH PROBABILITY OF DETECTION IS 0.5 OR GREATER

Confidence Rating	Processor	
	SLR	Non-SLR
Probable	20	20
Possible	24	24
Positive	25	26

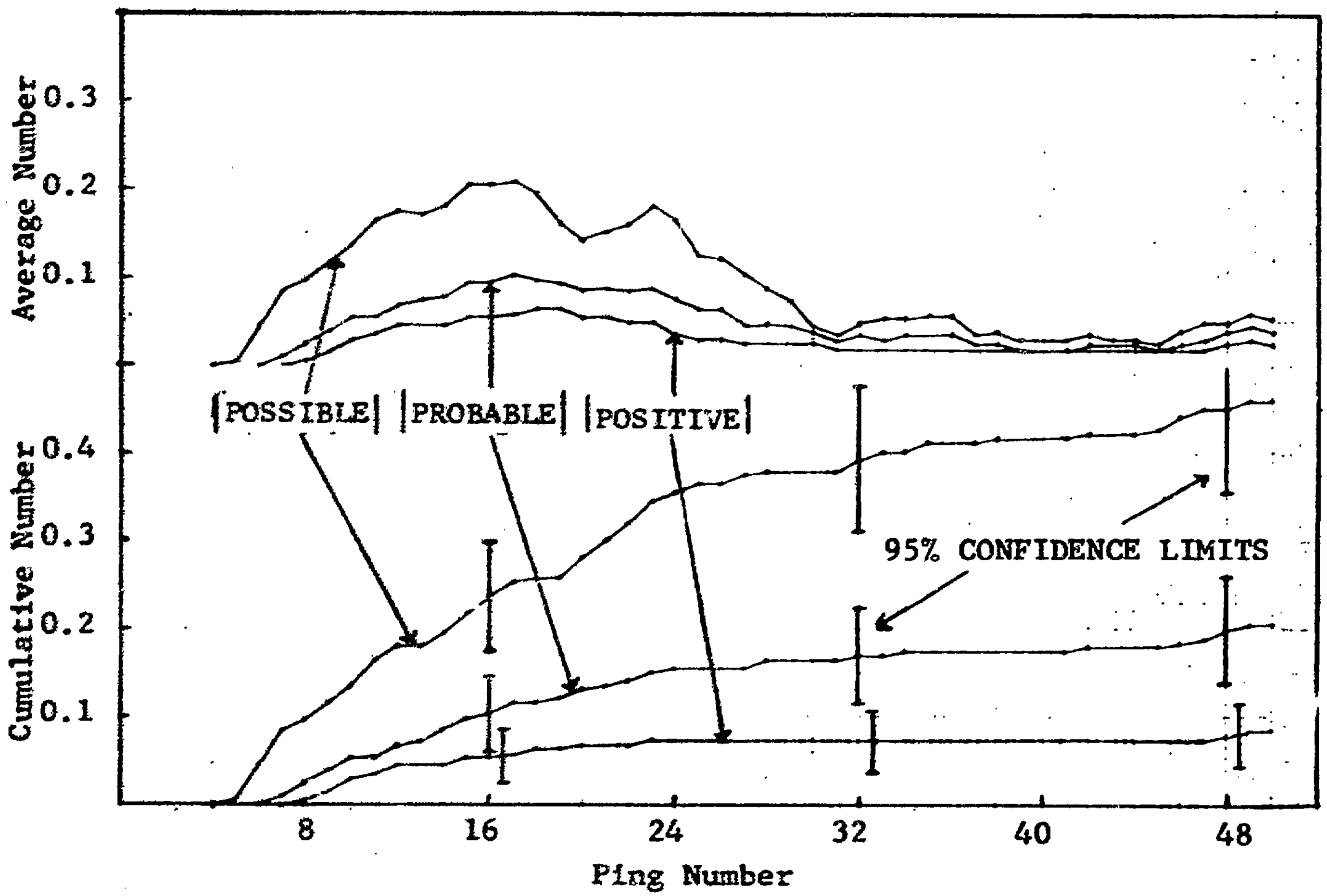


FIG. 5-9 - FALSE TARGET CALLS FOR SLR PROCESSOR AT THREE CONFIDENCE RATINGS, RUNS 1-12.

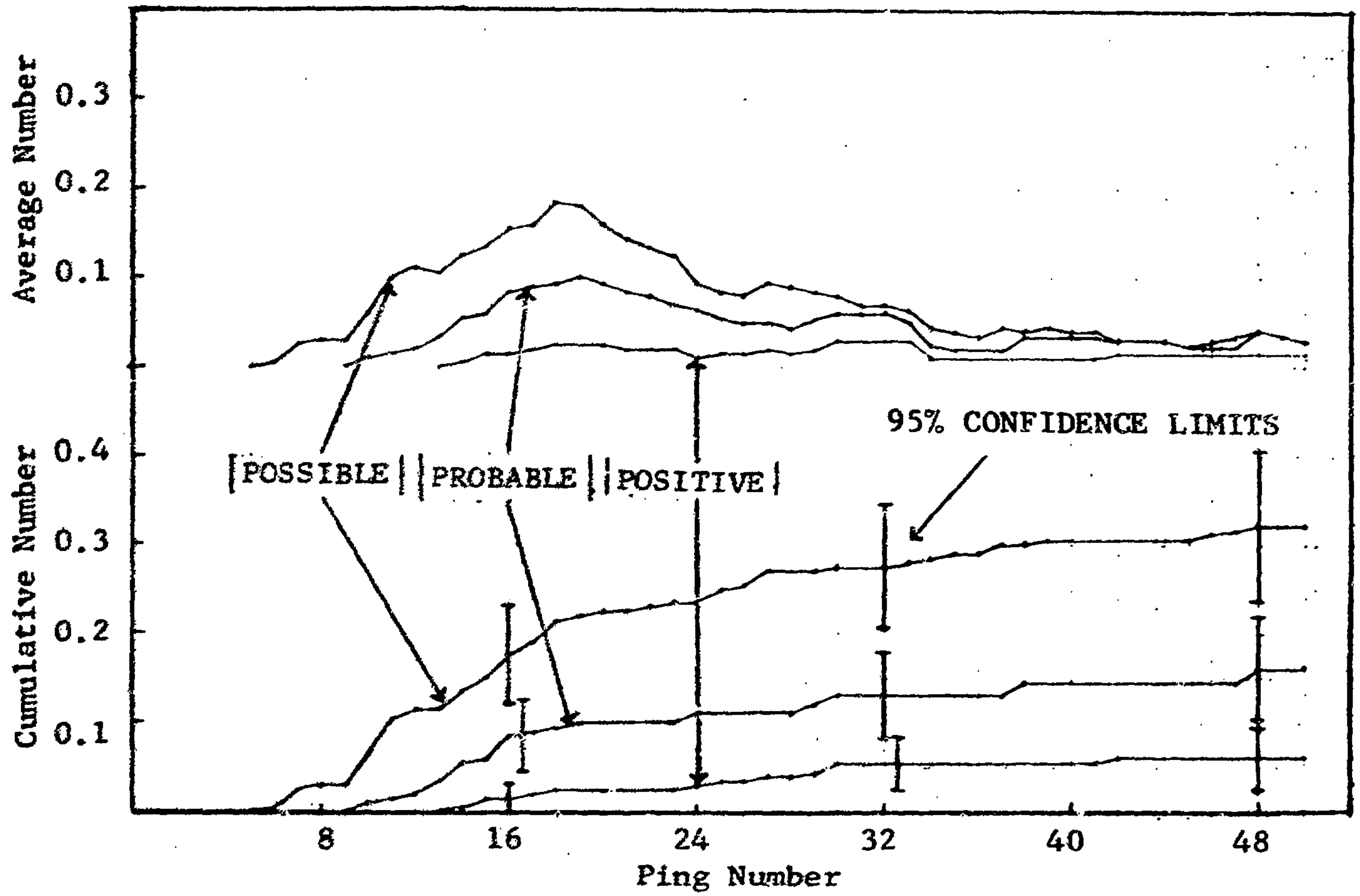


FIG. 5-10 - FALSE TARGET CALLS FOR NON-SLR PROCESSOR AT THREE CONFIDENCE RATINGS, RUNS 1-12.

average number of false targets called during a run and dividing by the total number of opportunities to call a false target. On a single ping basis, the number is simply the total number of spots on the grid that could be marked, 333 x 24 = 7992, minus those spots which would contain the true target track. Estimating the target track to be contained in five consecutive range bins and three consecutive bearing sectors, we have 7992 - 15 = 7977 opportunities to call a false target on each ping, or 50 x 7977 = 398,850 opportunities on any given run.* Reading from the bottom curves from Fig. 5-9, and from Fig. 5-10 which presents the same information for the non-SLR processor, false alarm rates for a given confidence rating are calculated and presented in Table 5-V.

Figures 5-11 and 5-12 present the false target responses to run 13 alone. Table 5-VI gives the estimated false alarm rates. It is difficult to draw conclusions from this data simply because the variability of subject responses is large enough to be a significant factor for sample sizes this small.

Figures 5-13 and 5-14 present false target calls obtained from three runs, runs 1, 4, and 9, with the true target track removed. Here we have 333 x 24 = 7992 opportunities to call a target on each ping, or 7992 x 50 = 399,600 opportunities on a given run. Table 5-VII gives the estimated false alarm rates for these runs.

Over runs 1-12 the observers recorded a total of 159 false target calls, or 159/405 = 0.39 false target calls per observer on the average. Looking only at runs 1, 4, and 9, 73 false target calls from a sample size of 104 were recorded, for an a an average of 73/104 = 0.70 per observer. The noise-only display

---

*This is the number that will be used to calculate false alarm rates for the five ping history.

TABLE 5-V

FALSE ALARM RATES FOR RUNS 1-12

Confidence Rating	Processor	
	SLR	Non-SLR
Possible	1.15×10^{-6}	0.807×10^{-6}
Probable	0.511×10^{-6}	0.404×10^{-6}
Positive	0.208×10^{-6}	0.150×10^{-6}

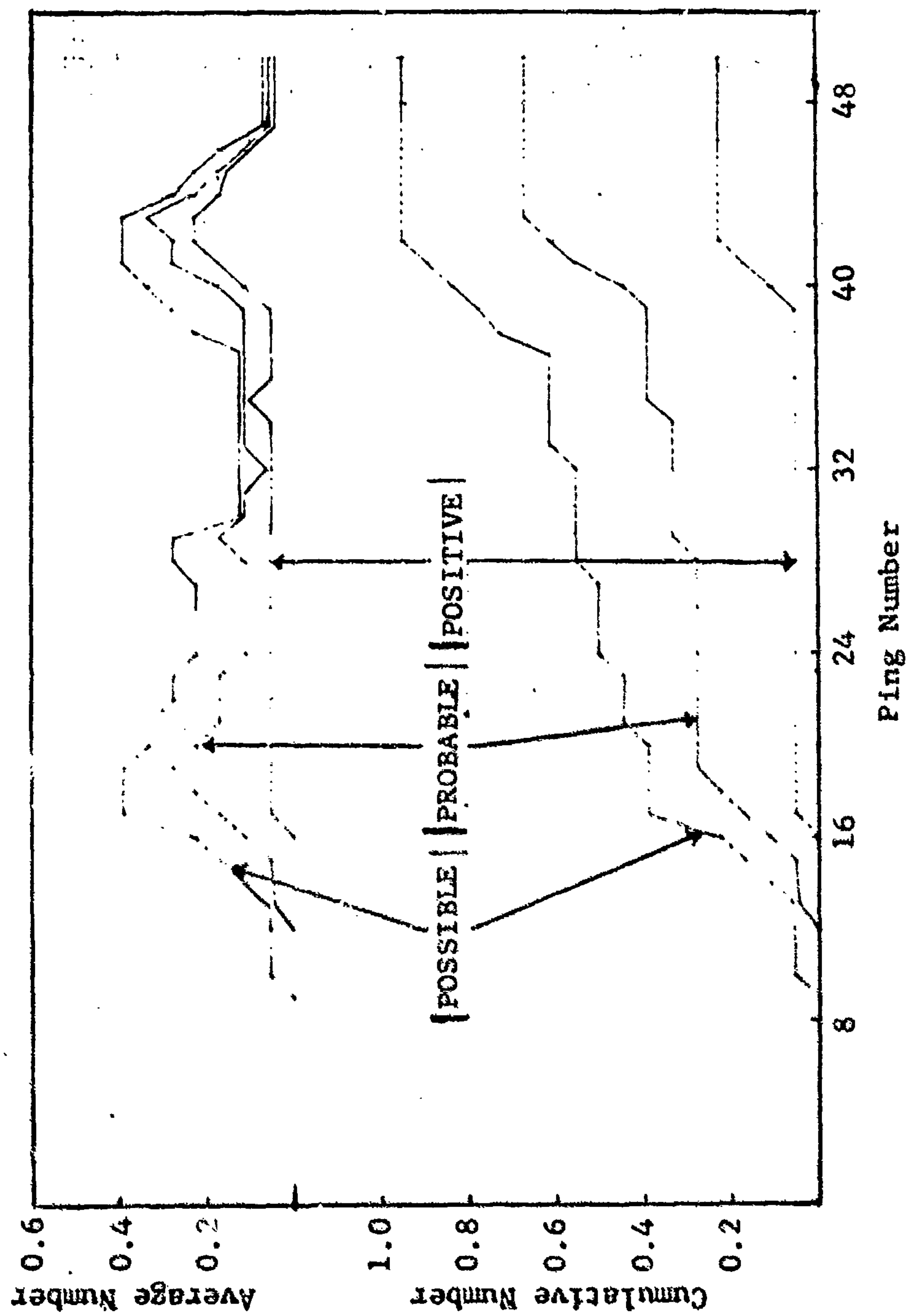


FIG. 5-11 - FALSE TARGET CALLS FOR SLR PROCESSOR AT THREE CONFIDENCE RATINGS, RUN 13

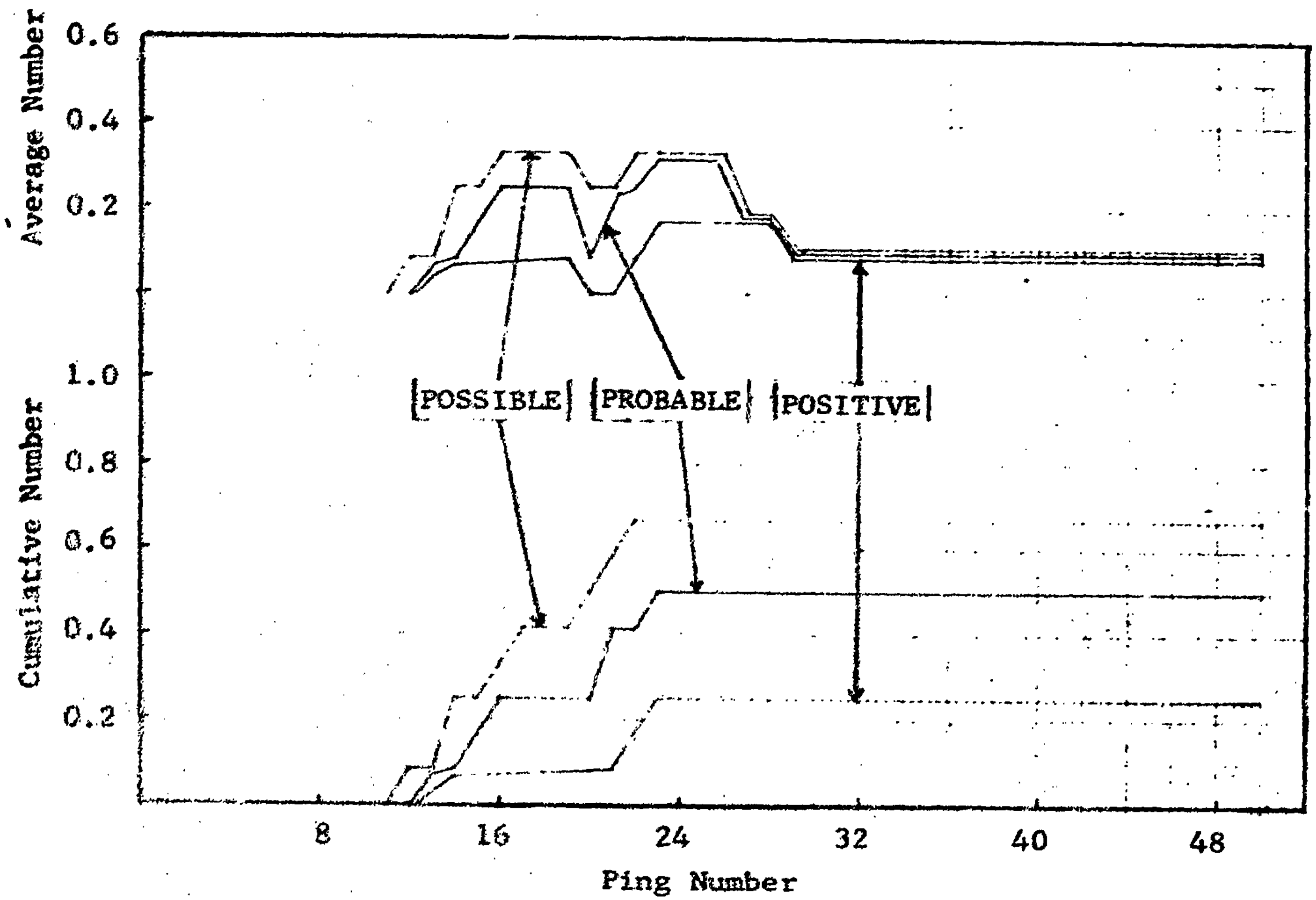


FIG. 5-12 - FALSE TARGET CALLS FOR NON-SLR PROCESSOR AT THREE CONFIDENCE RATINGS, RUN 13

TABLE 5-VI

ESTIMATED FALSE ALARM RATES FOR RUN 13

Confidence Rating	Processor	
	SLR	Non-SLR
Possible	2.37×10^{-6}	1.67×10^{-6}
Probable	1.67×10^{-6}	1.25×10^{-6}
Positive	0.557×10^{-6}	0.627×10^{-6}

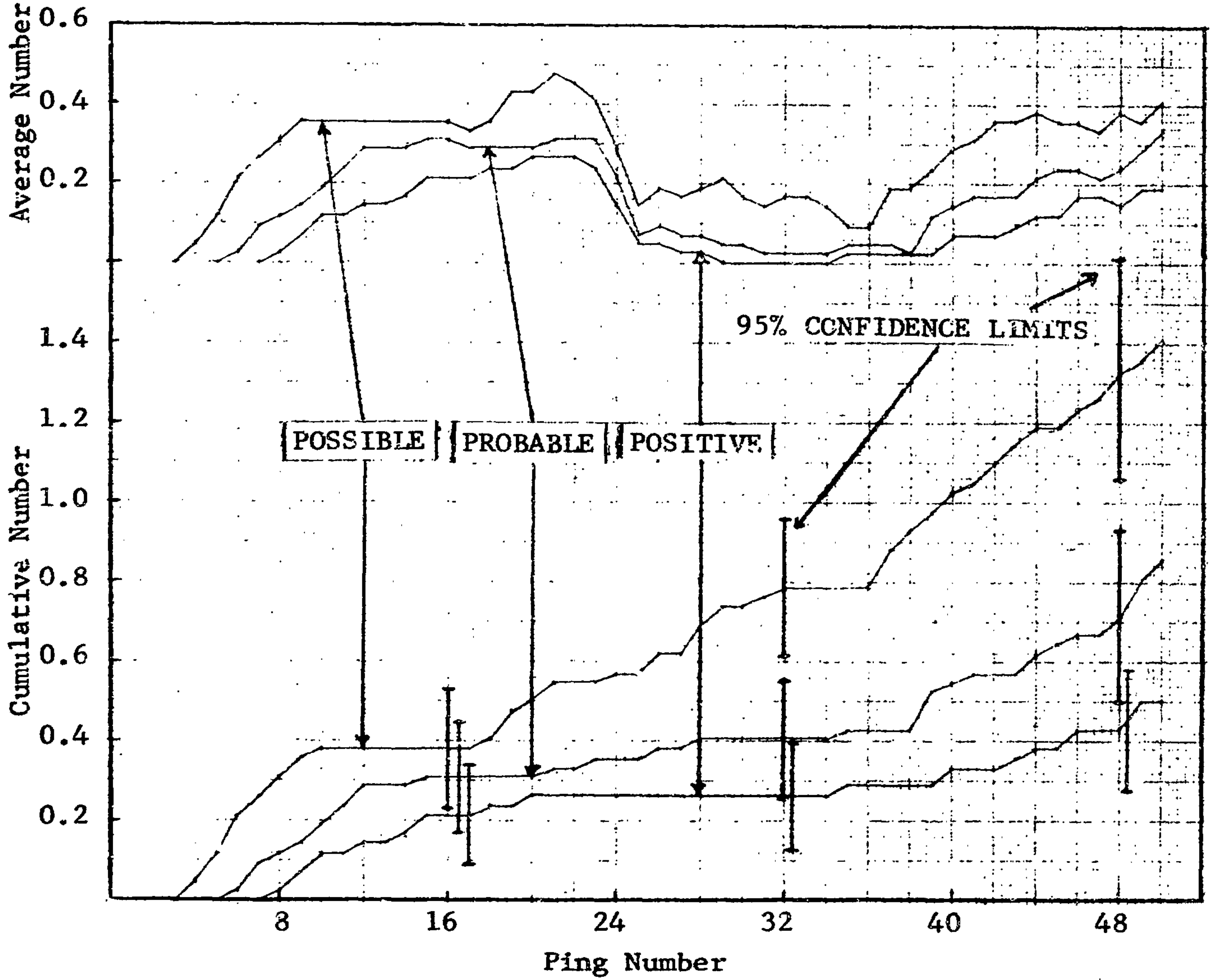


FIG. 5-13 - FALSE TARGET CALLS FOR SLR PROCESSOR AT THREE CONFIDENCE RATINGS, RUNS 1, 4, AND 9

TRACOR

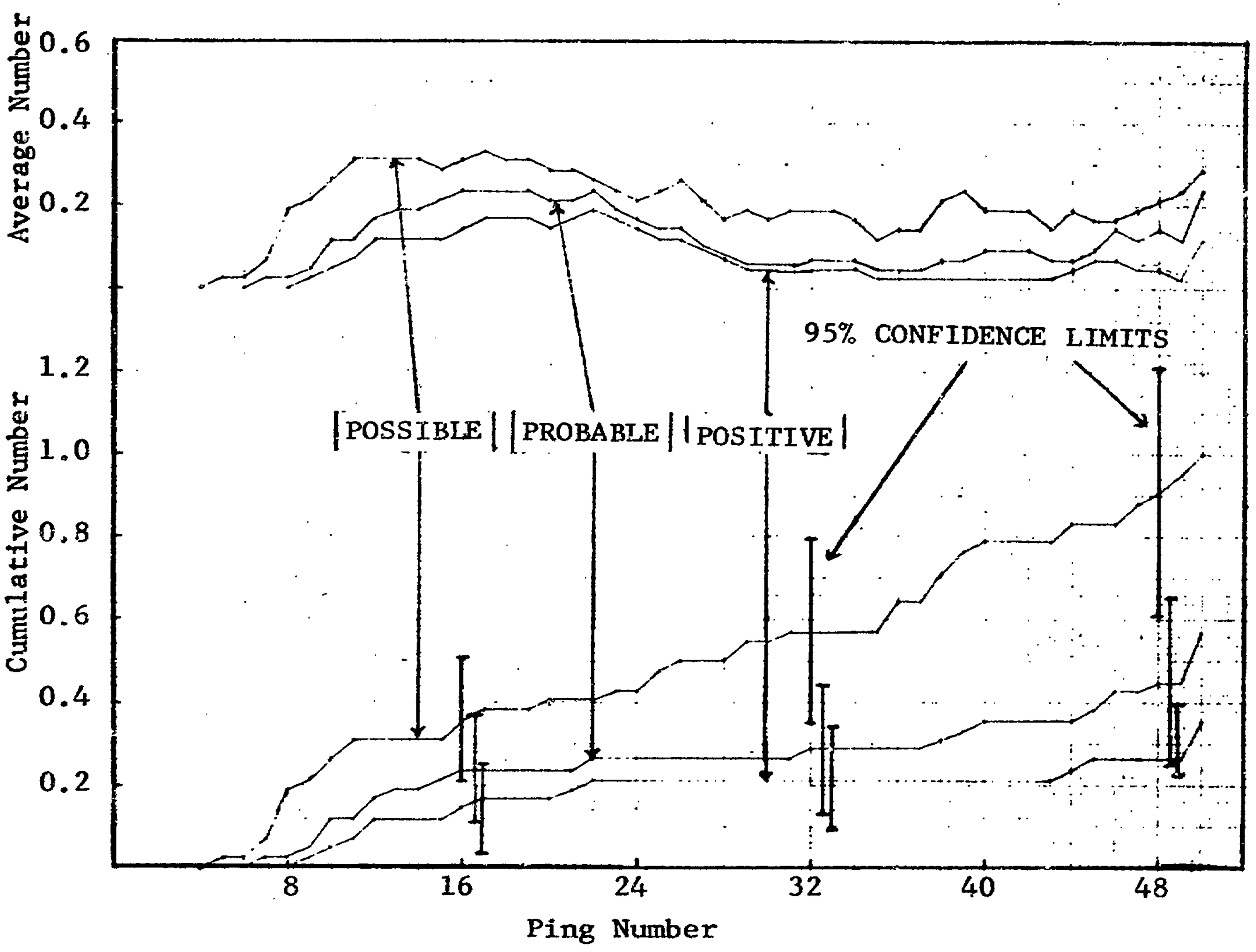


FIG. 5-14 - FALSE TARGET CALLS FOR NON-SLR PROCESSOR AT THREE CONFIDENCE RATINGS, RUNS 1, 4, AND 9

TABLE 5-VII

ESTIMATED FALSE ALARM RATES FOR RUNS 1, 4, AND 9
WITH TARGET TRACKS REMOVED

Confidence Rating	Processor	
	SLR	Non-SLR
Possible	3.52×10^{-6}	2.50×10^{-6}
Probable	2.14×10^{-6}	1.43×10^{-6}
Positive	1.25×10^{-6}	0.893×10^{-6}

from runs 1, 4, and 9, generated 101 false target responses from a sample size of 84, or 101/84 = 1.20 false targets per observer. The point is simply that as the target S/N increased, causing the target track to be marked at higher intensities than the background noise, the observers realized that this was "the" target and tended to stop calling any other potential targets.

An effort was made to measure the average target S/N level required for detection.* This result was obtained for each processor in the following manner: for a given confidence rating and for each display run, the first ping at which the true target track was identified was recorded. This ping number was then mapped with the proper S/N level defining the target track as is shown in Table 5-VIII. In the event a subject had not correctly identified the target by the end of the run (ping 50), a value of 20 dB was assigned. The S/N levels thus obtained were averaged and the results for each processor are given in Table 5-IX.

Table 5-X provides a comparison of several factors that can be obtained from the results. Each comparison is based on detection performance as measured by the earliest ping to correctly identify the target with a particular confidence rating. In each case, the results are broken out only into the two factors under consideration, providing each factor with approximately half the total sample size of 405.*

The first comparison gives the overall detection performance for the two processors. This is about the same result as given in Table 5-IV for the 0.5 probability of detection for each processor. It can be seen that there is really little difference in the two processors, given this particular data base.

---

*Runs 1-12 only.

TABLE 5-VIII

AVERAGE TARGET S/N LEVEL AT EACH PING

Ping No.	Average Target S/N Level (dB)
1-5	0
6-10	2
11-15	4
16-20	6
21-25	8
26-30	10
31-35	12
36-40	14
41-45	16
46-50	18

TABLE 5-IX

AVERAGE TARGET S/N LEVEL REQUIRED FOR DETECTION

Confidence Rating	Processor	
	SLR	Non-SLR
Probable	7.34	7.54
Possible	8.63	8.94
Positive	9.65	9.95

TABLE 5-X

SELECTED COMPARISONS BASED UPON AVERAGE EARLIEST PING NUMBER TO DETECT

Confidence Rating	Processor Type	
	SLR	Non-SLR
Possible	20.30	20.76
Probable	23.61	24.39
Positive	26.14	26.80
	Threshold Type	
	Fixed	Adaptive
Possible	20.50	20.54
Probable	24.17	23.80
Positive	26.58	26.34
	Background Type	
	High Density	Low Density
Possible	22.62	18.47
Probable	25.76	22.26
Positive	28.39	24.58
	Learning Type	
	Initial	Advanced
Possible	20.77	20.28
Probable	24.49	23.51
Positive	27.06	25.89

The next comparison gives a result that was not entirely anticipated. Initially it was thought that thresholding on a localized or adaptive basis would give better run to run performance than using a set of fixed thresholds for each run. This proved not to be the case. As to a comparison of the thresholding methods regarding false alarm rates, it was felt that due to the widely varying performance of the operators in calling false targets, no valid conclusions could be reached with these sample sizes. Therefore, the results from fixed and adaptive thresholding were combined as previously stated.

The third comparison gives a measure of performance regarding background densities. As mentioned in Section 5.5, the runs were divided into high and low density background runs for the purposes of randomizing the viewing sequence. Specifically, runs 1, 2, 3, 6, 10, and 11 were considered to have high density backgrounds, with an average of 10623 data points per run*, while runs 4, 5, 7, 8, 9, and 12 averaged only 2508 data points per run. As would be expected, it is easier to detect a strong target track in a low density background clutter, than in a high density background, all other things being equal.

The final comparison attempts to measure the observer learning curve, i.e., as they became more familiar with the display formats, background data, etc., the manner in which their detection performance varies. This table was generated by considering runs shown in the first half of the sequence vs. those shown in the second half of the sequence. It would appear that as the sequence progressed, a slight improvement in classifying the target track was shown.

---

*For the SLR processor.

Figures 5-15 and 5-16 present the results of an analysis of the behavior of each processor with respect to intensity levels associated with the target tracks in runs 1-12. These figures were generated by sorting each run ping by ping and recording the first ping for each run that a given threshold was was exceeded. The sorting routine made use of a dynamic target window extraction algorithm, which was developed due to some uncertainty in defining the location of the target tracks. Figure 5-17 present the average intensity levels per ping maintained by the target tracks, for each processor. Figure 5-18 gives the average track intensity level maintained per run for each processor.

5.7 Data Normalization

A comparative investigation of normalization techniques was not undertaken for this study. However, it appears from viewing the displays that work in this area may be necessary if a valid comparison of the two processors is to be made.

On practically every run there was a decided nonuniformity of marking with respect to bearing sectors. That is, certain sectors contained a high density of marks while adjacent sectors contained few, if any marks. This problem could possibly be remedied by normalizing within bearing sectors. Another problem with the data as observed was the appearance of a second target-like marking structure on several runs. These, of course, were called as targets by most of the subjects, and affected the false alarm rates. This problem would tend to degrade the overall performance of the SLR processor due to its tendency to integrate strong target signals rapidly.

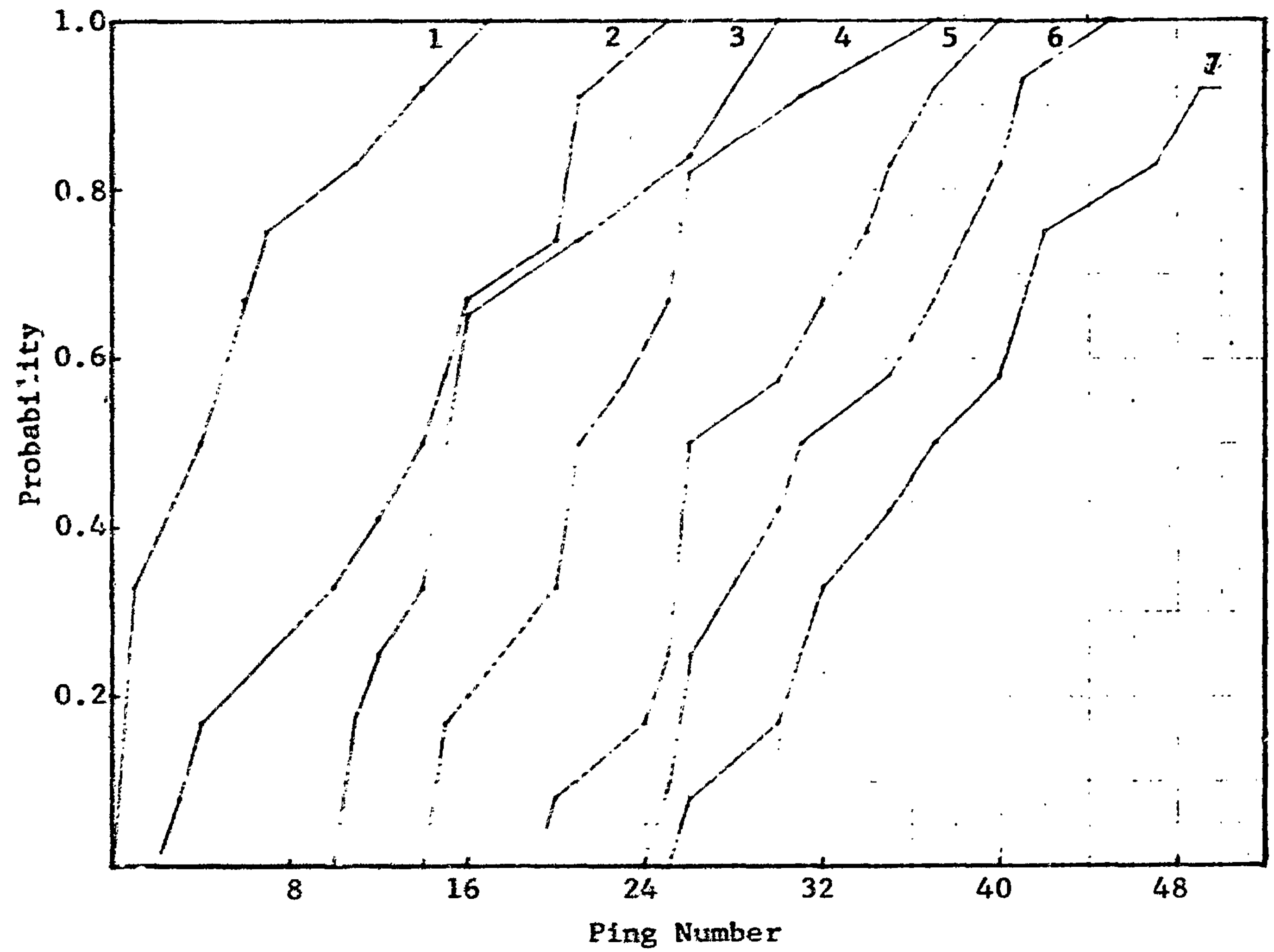


FIG. 5-15 - PROBABILITY OF CROSSING DISPLAY THRESHOLD FOR EACH THRESHOLD FOR SLR PROCESSOR (ENSEMBLE)

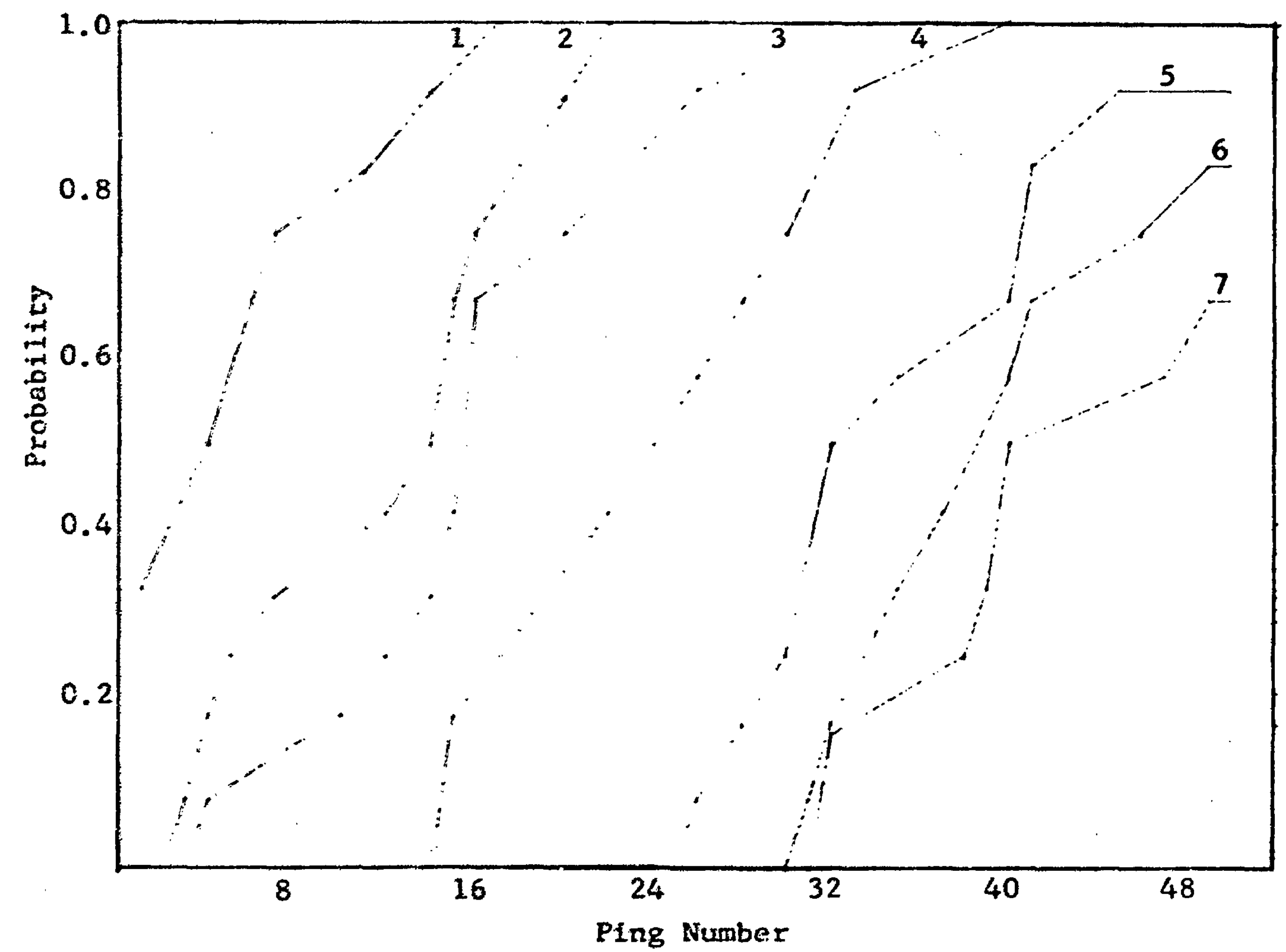


FIG. 5-16 - PROBABILITY OF CROSSING DISPLAY THRESHOLD FOR EACH THRESHOLD FOR NON-SLR PROCESSOR (ENSEMBLE)

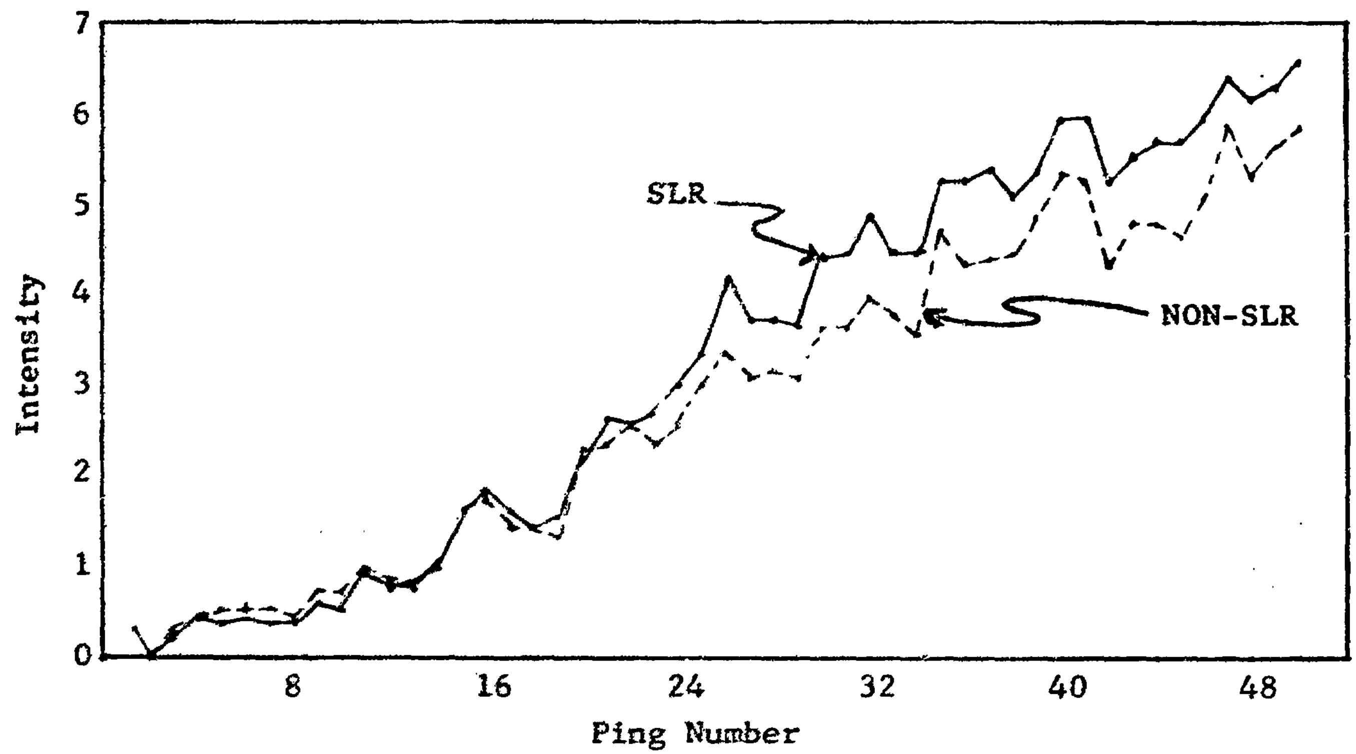


FIG. 5-17 - AVERAGE TARGET TRACK INTENSITY LEVEL FOR EACH PROCESSOR (ENSEMBLE)

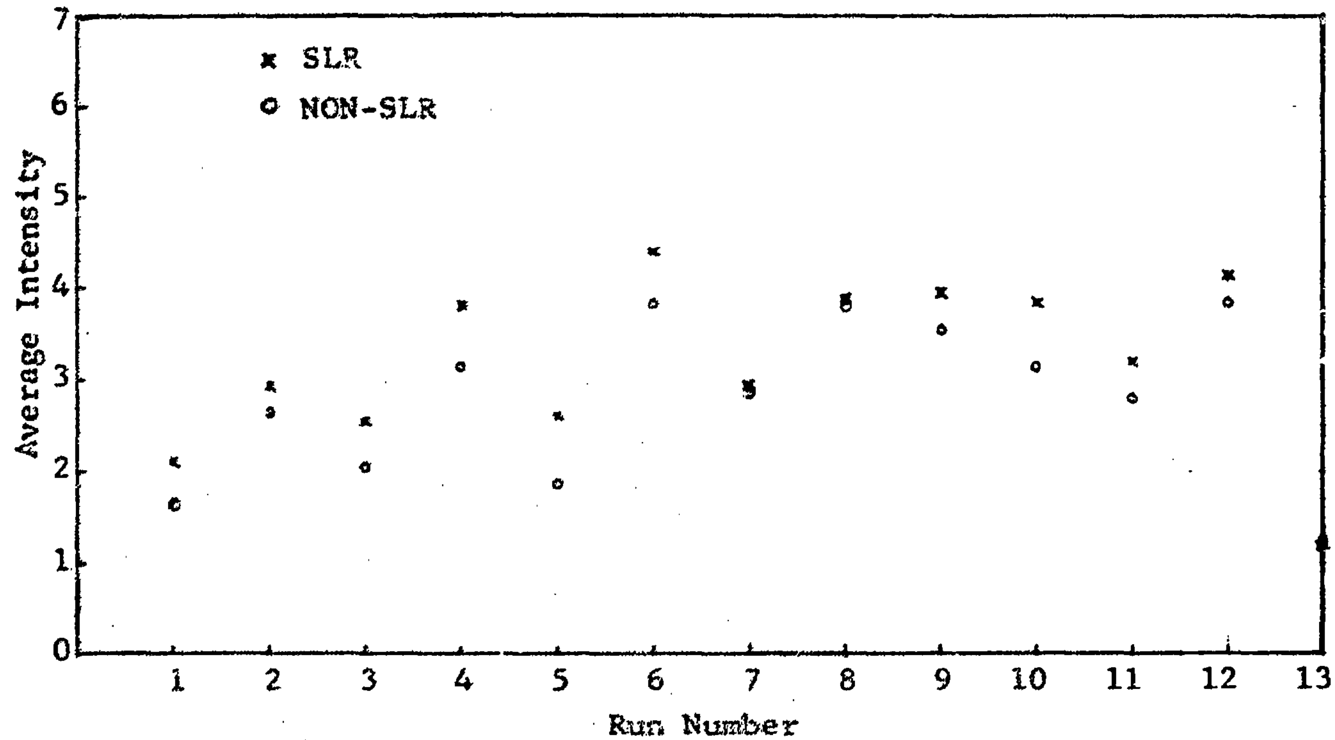


FIG. 5-18 - AVERAGE TARGET TRACK INTENSITY LEVEL FOR EACH PROCESSOR, BY RUN

5.8 Summary and Conclusions of Observer Study

A display study was performed using student observers trained to detect targets from a background of noise. Specifically, the study was carried out to validate the performance of TRACOR's Sequential Likelihood Ratio (SLR) processor using recorded multi-beam sea data. The Applied Research Laboratory (ARL) of the University of Texas supplied the data which consisted of stave recordings processed through a digital beamformer and signal processor, the outputs of which were made available on magnetic tapes in the form of 13 fifty-ping sequences or runs. A simulated target track was added to each run since the recorded data did not contain target signals of a nature that would provide meaningful processor comparisons.

TRACOR's CDC 3200 FORTRAN version of the SLR processor was used to process each run, ping by ping, obtaining as output on magnetic tape the significant range-bearing-log likelihood ratio triplets. These in turn were used to generate the displays in a range vs bearing format using a UNIVAC 1108 FORTRAN driver program and previously developed software. The entire viewing sequence, randomized with respect to background densities and processor types, was shown to the observers using TRACOR's black and white display facility consisting of 8 separate viewing stations or monitors. The tabulated observer responses were then used to obtain ping by ping summaries of correct and false target calls to be used in the processor comparisons.

The significant results of the study are listed as follows:

(1) There was no appreciable difference in detection performance between the SLR and non-SLR processors. Detection performance was measured by the number of correct calls

identifying the target out of the total number of opportunities to identify the target.

(2) The false alarm rates were somewhat lower for the non-SLR processor, especially for the noise-only runs. The false alarm rates were determined by dividing the number of false alarms by the total number of opportunities to make false alarms.

(3) Average target track intensity levels as viewed on the displays were somewhat higher (brighter) for the SLR processor.

(4) Relatively small variations in setting the display brightening thresholds had no effect upon the detection performance of either processor.

(5) Observer responses to false targets were greatly reduced in the presence of a strong target signal for either processor.

The study was somewhat limited in scope in that the encounters investigated assure fairly rapid and certain detectability independent of the processor being considered. Additional investigations should define a large sampling of encounters that would demonstrate marginal detection performance for each processor.

6.0 RECOMMENDATIONS

The SLR processor has progressed from a relatively crude one-dimensional processor into a sophisticated multidimensional, multichannel processor capable of processing large amounts of varied data types. The SLR processing of sea data for the observer test points out one of the problem areas for implementation--the necessity of normalization. A basic underlying assumption in the SLR processor is that the input data is stationary in time and space. For the particular data base used in the present study this was not true due to noise spokes at certain bearings and variations in noise backgrounds from run to run. Based on this experience, it would be well to look into how sensitive the SLR processor is to nonstationarity. The results of this study, along with existing information on how well various normalization schemes work, would allow the proper selection of normalization procedures.

When the SLR processor is considered for implementation on a moving platform, problems will arise in matching previously stored data with incoming data. There are two possible ways of handling this problem. The first establishes the ship as the basic reference point and allows targets to move relative to the ship. The second establishes a geographic point on the earth's surface as the basic reference point, and the ship and targets move in a fixed grid. The various trade-offs between the two systems should be investigated in light of the SLR processor and computer requirements.

Previous studies* indicate that an improvement in the tracking algorithm may yield significant gains in processor

---

*H. A. Reeder, "Simultaneous Likelihood Ratio Processing for Two Active Receivers," Vol. I, TRACOR Document T71-AU-9594-U, 25 August 1971.

performance. One possible way of doing this is to utilize observed tracking errors (difference between predicted and observed target locations) in the log likelihood ratio. Using this concept, widely varying tracks would be penalized much more than consistent tracks. The tracking algorithm and its effect on detection performance and computer loading is potentially the most fruitful area for improving the individual SLR processors.

The results of the recent observer study were disappointing in that they did not show significant differences between SLR and non-SLR processing. However, the type of data and the amount of data were not sufficient to make a definitive study. If a different, more extensive data base is available, it would be worthwhile to undertake a carefully controlled observer experiment.

The SLR processors for the Multireceiver Automatic Alerting System are nearly complete, lacking only the narrowband processor. It is recommended that this SLR processor be implemented on a representative narrowband sonar signal processor and evaluated. Also a joint wideband, narrowband SLR processor should be developed and tested. The ultimate aim of this combined system should be hardware implementation on an at-sea platform for definitive operational testing.

7.0 BIBLIOGRAPHY

1. H. A. Reeder, "Computer Utilization of Sequential Hypothesis Testing for Detection and Classification of Sonar Signals," TRACOR Document 67-717-U, 27 October 1967, (UNCLASSIFIED).

2. H. A. Reeder, "Estimation of Computer Requirements for a Detection Technique Based on Sequential Hypothesis Testing," TRACOR Document 68-352-U, 1 March 1968, (UNCLASSIFIED).

3. H. A. Reeder, L. L. Cortes and H. D. Record, "Detailed Feasibility Study of a Sequential Likelihood Ratio Processor for the AN/SQQ-23 (PAIR) Sonar," TRACOR Document 68-869-U, 29 July 1968, (INCLASSIFIED).

4. H. A. Reeder and H. D. Record, "Summary Report-Computer-Aided Detection (U)," TRACOR Document 68-1238-C, 19 November 1968, (CONFIDENTIAL).

5. H. A. Reeder, "Reduction of Computer Requirements for the Sequential Likelihood Ratio Processor," TRACOR Document T70-AU-7242-U, (UNCLASSIFIED).

6. H. A. Reeder, "Simultaneous Likelihood Ratio Processing for Two Active Receivers," Vols. I and II, TRACOR Document T71-AU-9594-U, 25 August 1971, (UNCLASSIFIED).

7. H. A. Reeder, "Computer Aided Detection for Wideband Passive Sonar Systems (U)," TRACOR Document T73-AU-9520-U, 14 February 1973, (UNCLASSIFIED).

APPENDIX A

MATHEMATICAL FORMULATION OF THE SLR PROCESSOR AND DESCRIPTION OF THE SLR COMPUTER PROCESS

A.1 STATISTICAL HYPOTHESIS TESTING AND THE LIKELIHOOD RATIO

A.1.1 Simple Alternative Binary Tests

This appendix serves two purposes. First, it gives the reader a brief introduction to statistical decision theory emphasizing the central role played by the likelihood ratio. Second, it describes the operation of the sequential likelihood ratio processor as it is currently implemented by TRACOR.

The fundamental problem of statistical decision theory is that of choosing one of several possible hypotheses by utilizing information gained from the measurement of some quantity. A great deal of generality can be included in defining the algorithm to optimally carry out this procedure, but a simplified approach will be taken here. For a background in statistical detection theory, the reader is referred to Helstrom[1] and Hancock and Wintz[2], and for further study in sequential analysis, the reader is referred to Wald.[3]

[1]Carl Helstrom, Statistical Theory of Signal Detection, Pergamon Press, New York, 1960.

[2]John C. Hancock and Paul A. Wintz, Signal Detection Theory, McGraw Hill Book Company, New York.

[3]A. Wald, Sequential Analysis, John Wiley & Sons, Inc., New York, 1948.

The problem now is to decide whether to accept H_0 or H_1 given the observation x where H_0 is the noise hypothesis and H_1 is the signal hypothesis. In the conceptually simple problem posed here we have a case of simple hypothesis testing which has been treated extensively in the literature. An excellent treatment of this problem is given in Ref. [2]. In this reference it is shown that a functional known as the likelihood ratio plays a central role in the theory of statistical hypothesis testing. The likelihood ratio L(x) is defined by,

$$L(x) \triangleq \frac{p_1(x)}{p_0(x)}$$

where $p_1(x)$ and $p_0(x)$ are the probability density functions associated with the hypothesis H_1 and H_0, respectively. The sense in which this quantity plays a central role is the following. When one attempts to make decisions of the type dealt with above, it is reasonable to attempt to make that decision in a fashion that is optimum according to some criterion. Through the years, there have been numerous criteria of optimality developed. The variation in criteria of optimality has resulted from essentially two causes. First, the situations or contexts in which the decision is to be made are diverse, *e.g.*, testing lots of seeds for the presence of certain strains or testing electrical signals for the presence of an echo from an attacking aircraft or guided missile. Clearly, the motivations and constraints behind these two decision processes are different. Second, the level of knowledge concerning the factors other than $p_0(x)$ and $p_1(x)$ is highly variable. For example, there are cases where the costs of incorrect decisions can be assessed fairly accurately as can be the *a priori* probabilities of the events associated with H_0 and H_1. Before enumerating some of the more commonly encountered criteria used in statistics it will be convenient to introduce some notation. When binary (yes-no) decisions are made, two types of errors can be made. The probability of these errors plays an important role in describing the utility of

the particular approach that is taken to decision making. Using conventional notation these errors are,

α = probability of rejecting H_0 when in fact H_0 is true.

This is called a type 1 error by statisticians but in a target detection theory context it amounts to the probability of rejecting the "target-absent" hypothesis when in fact the target is absent. Thus, this is the <u>false</u> alarm probability.

β = probability of rejecting H_1 when H_1 is true.

This is called a type 2 error and in the target detection sense is the probability of rejecting the "target-present" hypothesis (H_1) when in fact the target is present. Thus, $1-\beta$ is the probability of detection.

Two other quantities of some importance are the *a priori* probabilities of H_0 and H_1, namely π_0 and π_1, respectively. Often in practice these quantities are not known to the observer and some of the criteria to be discussed shortly will reflect this fact. Finally, there are costs that may sometimes be associated with the four possible outcomes of the decision process. These are;

C_{00} = cost associated with choosing H_0 when H_0 is true

C_{01} = cost associated with choosing H_0 when H_1 is true

C_{11} = cost associated with choosing H_1 when H_1 is true

C_{10} = cost associated with choosing H_1 when H_0 is true.

Having defined the basic quantities associated with simple binary hypothesis testing, some of the more common decision theory criteria of optimality will be briefly discussed. Let it be borne in mind, however, as these criteria are discussed, that the function of a criterion is to tell us what must be done with the observation in order that the criterion be satisfied. This is simply finding the operation that must be performed on the data. Table A-I gives the criteria and the properties associated with each. As can be seen in this table, regardless of the choice of criterion, the common operation that must be performed on the data is the formation of the likelihood ratio. Only the threshold testing procedure is dependent on the state of a priori knowledge and the motivation of the observer. Thus, it is clear that the likelihood ratio is a most fundamental operation in statistical hypothesis testing. In the text that follows, our implementation of the likelihood ratio and the subsequent application to processing sonar data will be discussed.

TABLE A-I

STATISTICAL DECISION CRITERIA AND THEIR PROPERTIES

Criterion	PROPERTIES: Description	Required Data Operation	*A priori* Information Required, i.e., π_0, π_1, C_{00}, C_{11}, C_{10}	Required Sample Size, i.e., Number of x's
Neyman-Pearson Observer	Minimizes β for a given α. (Fixes the false alarm prob. and maximizes prob. of detection.	Form L(x) and test against threshold T that gives desired value of α.	None	Fixed
Ideal Observer	Minimizes the average probability of error $P_E=\pi_0\alpha+\pi_1\beta$.	Form L(x) and test against threshold T that is dependent on π_0 and π_1.	π_0 and π_1	Fixed
Sequential Observer	Minimizes the average number of samples required to give a specified α and β.	Form L(x) and test against two thresholds which are determined by α, β, and in some cases π_0 and π_1 when available.	π_0 and π_1 when available	Random

TABLE A-I

STATISTICAL DECISION CRITERIA AND THEIR PROPERTIES (Cont'd)

Criterion	PROPERTIES			
	Description	Required Data Operation	*A priori* Information Required, *i.e.*, π_0, π_1, C_{00}, C_{11}, C_{10}	Required Sample Size, *i.e.*, Number of x's
Bayesian Observer	Minimizes the average risk, R. $R=\pi_0 C_{00}(1-\alpha)$ $+\pi_1 C_{11}(1-\beta)$ $+\pi_0 C_{10}\alpha+\pi_1 C_{01}\beta$	Form L(x) and test against threshold T determined by π_0, π_1 and the costs.	π_0, π_1, C_{00}, C_{11}, C_{10}.	Fixed
Minimax	Minimizes the average risk, R, after maximizing R with respect to π_0.	Form L(x) and test against threshold that is determined by the costs and value of π_0 that maximizes R.	C_{00}, C_{01}, C_{11}, C_{10}.	Fixed

In the present analysis it is assumed that the quantity to be observed is a single numerical quantity (voltage) available at the output of the sonar processor (e.g., a correlator). This quantity is indexed by ping number, range, bearing, and possibly Doppler. In other words we have available amplitude information from every range-bearing-Doppler resolution cell, on every ping cycle. If the hypothesis H_0 is true for the particular elemental volume of the ocean in question, then observed values of the quantity x will be described by a known probability density function, $p_0(x)$, such as the example shown in Fig. A-1. Similarly, if the hypothesis H_1 is true, there will be a different probability density function, $p_1(x)$, which describes the quantity x, as shown in Fig. A-1. For this example, $p_0(x)$ is the probability density function describing the processor output with noise alone input to the processor, and $p_1(x)$ is the probability density function describing the processor output with signal-plus-noise input to the processor. Thus, the likelihood ratio is defined as before by,

$$L(x) \triangleq \frac{p_1(x)}{p_0(x)} .$$

In many cases and as shown in Fig. A-1, the likelihood ratio is a monotonically increasing function of the observation quantity x. Here, large values of L(x) tend to imply that H_0 is true. Thus, in principle, a decision threshold can be established for testing L(x) according to one of the criteria presented in Table A-I

A.1.2. Multiple Observations

If n observations of the quantity x are to be made at separate points in time, or specifically on successive echo cycles, thereby resulting in the sequence $(x_1, x_2, x_3, \ldots, x_n)$

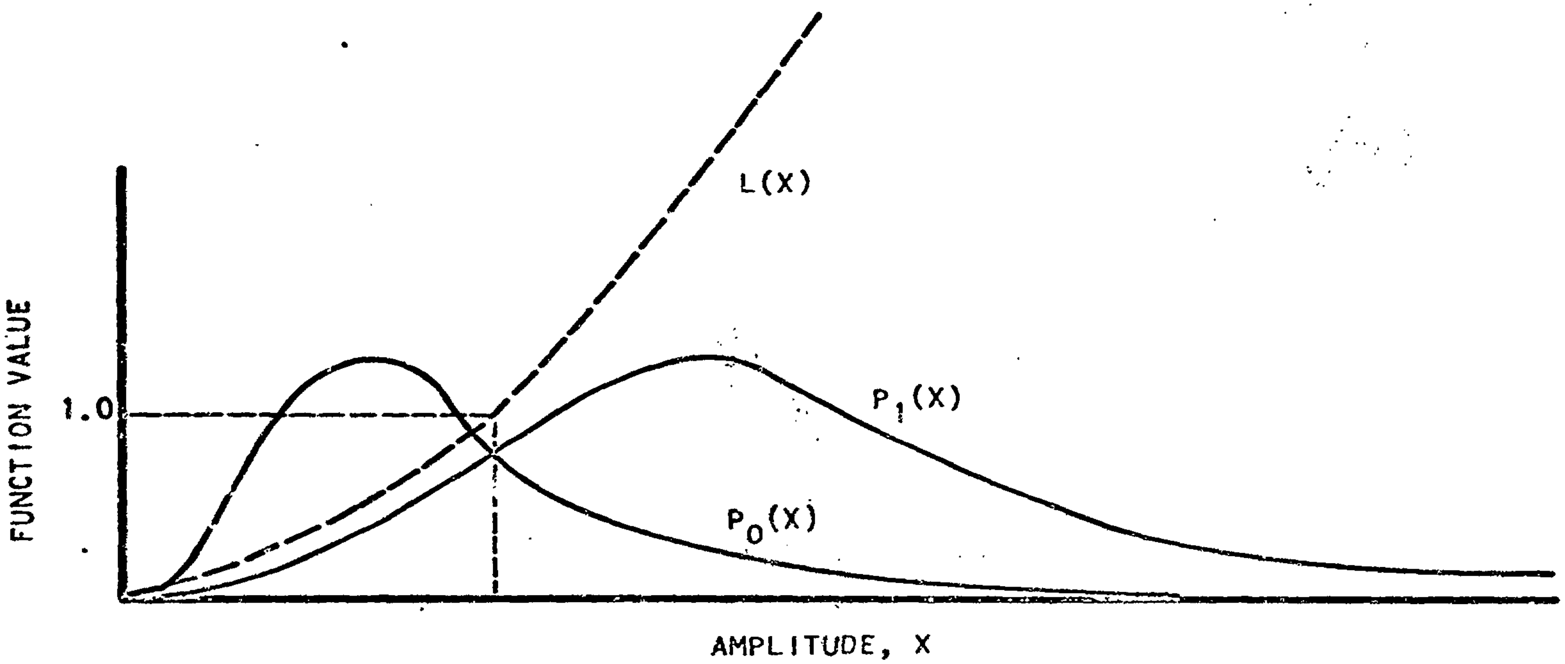


FIG. A-1 TYPICAL PROBABILITY DENSITY FUNCTIONS AND LIKELIHOOD RATIO

then a joint likelihood ratio, $L(x_1, x_2, x_3, \ldots, x_n)$ can be defined based upon the n dimensional probability density functions, $p_1(x_1, x_2, x_3, \ldots, x_n)$ and $p_o(x_1, x_2, x_3, \ldots, x_n)$, similar to $p_o(x)$ and $p_1(x)$. The joint likelihood ratio is then

$$L(x_1, x_2, x_3, \ldots, x_n) \triangleq \frac{p_1(x_1, x_2, x_3, \ldots, x_n)}{p_o(x_1, x_2, x_3, \ldots, x_n)}.$$

If the observations $(x_1, x_2, x_3, \ldots, x_n)$ can be considered statistically independent, then the appropriate multidimensional probability density function can be described as the product of the individual probability density functions, thus

$$L(x_1, x_2, x_3, \ldots, x_n) = \frac{p_1(x_1) \cdot p_1(x_2) \cdot p_1(x_3) \ldots \cdot p_1(x_n)}{p_o(x_1) \cdot p_o(x_2) \cdot p_o(x_3) \ldots \cdot p_o(x_n)}.$$

This yields a significant simplification in the determination of processor output statistics, and leads to the suggestion of the log likelihood ratio, $\ell(x_i)$, which is formed by taking the logarithm of $L(x_i)$, thus

$$\ell(x_i) \triangleq \text{Log}\left[L(x_i)\right] = \text{Log}\left[\frac{p_1(e_i)}{p_o(x_i)}\right],$$

$$\ell(x_i) = \text{Log}\left[p_1(x_i)\right] - \text{Log}\left[p_o(x_i)\right], \text{ and}$$

$$\ell(x_1, x_2, x_3, \ldots) = \ell(x_1) + \ell(x_2) + \ell(x_3) + \ldots, + \ell(x_n).$$

The procedure of adding rather than multiplying lends itself quite well to a digital computer, however, the process of taking a logarithm can be time consuming. Consequently, a linear approximation to the log likelihood ratio deserves consideration. This will be discussed in a later section of this appendix.

A.1.3 Sequential Hypothesis Testing

It is of interest to consider next a system in which the number of observations is not a fixed quantity but, instead, a decision is to be made when specified confidence levels are reached. This technique which was introduced earlier, is known as sequential testing, and requires that two thresholds, T_L and T_D be established. In the case at hand, the threshold T_L, with $(\underline{x}) = (x_i, x_{i+1}, \ldots, x_{j-1}, x_j)$, is chosen such that if the value of the log likelihood ratio, $\ell(\underline{x})$, falls below T_L, the decision is made that H_0 is true, that no target is present. Thus, the track is rejected as noise, and the testing chain stops.

Similarly, T_D is chosen such that if $\ell(x)$ exceeds T_D, the decision is made that H_1 is true, that a target is present. This completes the detection process in a sense, but in our application the testing procedure does not stop. Rather, the sequential testing continues and forms an automatic track. If the value of $\ell(x)$ lies between the thresholds, that is, if

$$T_L < \ell(\underline{x}) < T_D$$

then the decision is made to retain the track, but not display it. Following this, another sample is taken, $\ell(\underline{x})$ is updated, and the new $\ell(\underline{x})$ is compared with T_L and T_D.

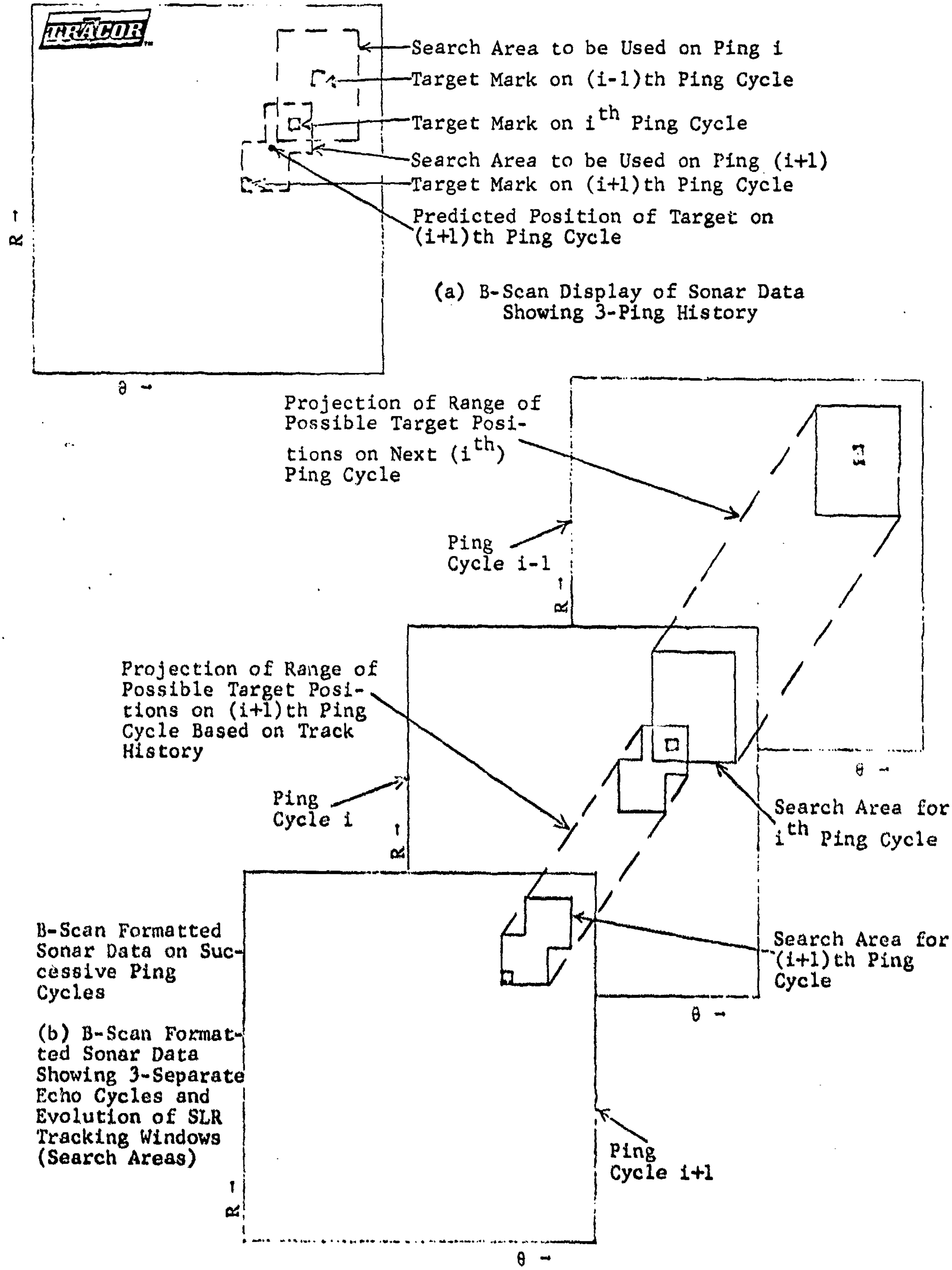
This process is very similar to the random walk problem, and it can be shown that eventually, with probability 1, one of the two thresholds will be crossed and a decision will be reached. The average number of samples required to reach a decision for given probabilities of wrong decisions for the sequential test described above, is less than the number required for a fixed sample-size test with the same probabilities of error, α and β*. If the track is noise, the decision is reached relatively promptly, but if the track is a target, a significantly greater number of samples may be required for a decision.

A.1.4 Tracking With the SLR Procedure

In sonar applications one difficulty arises which does not often occur in other statistical decision theory applications. This problem is that one does not really know uniquely how to make a single "next" observation. For example, receipt of a modestly large sample on one beam, at a given range and perhaps with some Doppler, gives an indication of approximately where to look in the next echo cycle, in terms of beam, range, and Doppler. This information, however, cannot give a precise specification of the location of the linking sample in the next echo cycle. Thus, the process is more complex than the classical sequential test which is conducted in a single resolution cell. The information, i.e., range, bearing, Doppler, and amplitude, gained from the received sample on the present ping cycle defines a range-bearing-Doppler volume in the next echo cycle which must be searched if data on successive ping cycles are to be associated or linked. This process can be better understood if one examines the situation depicted in Fig. A-2.

In this figure, the upper diagram (Fig. A-2(a)) shows a highly idealized version of a target track that has developed over a three ping sequence. On the $(i-1)^{th}$ ping the

*A. Wald, op. cit.



(a) B-Scan Display of Sonar Data Showing 3-Ping History

(b) B-Scan Formatted Sonar Data Showing 3-Separate Echo Cycles and Evolution of SLR Tracking Windows (Search Areas)

FIG. A-2 - DIAGRAMATIC REPRESENTATION OF SLR ALGORITHM TRACKING DYNAMICS

target occupies a certain range-bearing resolution cell. Since this is the only information (i.e., range, bearing, and amplitude) that we have concerning the target, it must be concluded that on the next ping, ping i, the target is likely to appear anywhere in a rather large area of range and bearing whose size is dictated by the maximum target motion that could occur between pings. This search area is shown in the upper right-hand part of Fig. A-2(b) as the large, heavily marked area around the target mark. As is shown in this figure, this search area is projected onto the B-scan format of the i^{th} echo cycle so that the search for a target mark to associate with the $(i-1)^{th}$ ping cycle target mark can be confined to a reasonable area. On the i^{th} ping cycle, we have shown the target mark appearing in the lower left-hand corner of the i^{th} ping cycle search area. Referring back to Fig. A-2(a), it can be seen that with the two pings of target motion history it is possible to predict a future target position. This is shown in Fig. A-2(b) as a projection of the irregular area from ping cycle i to ping cycle i+1. It can be seen that the area covered by the search area for ping i+1 is smaller than the search area that was used for the i^{th} ping cycle. This is a reasonable result in view of the fact that more information is available at the end of the i^{th} ping cycle than was available at the end of the $(i-1)^{th}$ ping cycle. (It is worth pointing out here that the combination of the information such as range, bearing, log likelihood ratio, predicted target position, and size of search area constitute what we call status unit.)

Now when the target echo appears in the projected search areas from one ping cycle to the next, the SLR tracking algorithm takes advantage of this by forming the joint log likelihood ratio that was discussed in the previous section. This gives an enhancement of the target echo and tends to increase the signal-to-noise ratio. Unfortunately, as the reader has probably already guessed, it is possible for two other types of events to

occur in this tracking procedure in addition to the pure enhancement of target tracks. First, there can result spurious noise tracks. These are tracks that develop after a target track has been initiated and when a noise-alone sample falls in the search area and is large enough to cause a branching track from the target track. This results from the fact that the algorithm, in its most general form, permits multiple linkages to be formed between old tracks and new data samples that fall in the search area. Second, it is entirely possible that noise alone can cause tracks to be initiated. This occurs whenever the noise sample exceeds the lower threshold and thereafter is treated in the same way as the target echo of Fig. A-2 was treated. Here again multiple linkages can result so that it is possible for a multiplicity of noise tracks to exist. The technique by which this noise behavior is controlled is discussed next.

Since multiple linkages are allowed in this process, the definition of the joint likelihood ratio must be altered slightly. The joint likelihood ratio for the case in which multiple linkages are allowed can now be defined as

$$L'(x) \triangleq \frac{p_1(\underline{x} \mid H_1)P_1}{p_0(\underline{x} \mid H_0)P_0} ,$$

where $p_i(\underline{x} \mid H_i)$ is the conditional probability density function of $\underline{x}$ given that hypothesis H_i is true. P_1 is the probability that a linkage formed is with the target, given that H_1 is true*; and P_0 is the probability that the linkage formed is due to noise, given that H_0 is true.

*If multiple linkages are allowed, there is a probability that a true target track can link with a noise sample, as well as a target sample, consequently the joint likelihood ratio must be modified by P_1.

For the purpose of this analysis, it is assumed that P_0 is exactly unity. It is also assumed that P_1 is the number of true target tracks within the search area or volume divided by the number of possible linkages, including target and noise linkages. It is unlikely that there will be more than one target track in any given search area when H_1 is chosen, so the average number of target tracks there is 1.0. Since it would be difficult and time consuming to count the number of possible linkages each time, the average number of possible linkages N_{AV}, will be used, hence

$$P_0 = 1.0, \text{ and}$$

$$P_1 = \frac{1.0}{N_{AV}} .$$

Thus the likelihood ratio $L(\underline{x})$ must divided by the average number of possible linkages under hypothesis H_1. That is,

$$L'(\underline{x}) = \frac{L(x)}{N_{AV}}, \text{ and}$$

$$\ell'(\underline{x}) = \ell(\underline{x}) - \text{Log } (N_{AV}).$$

This reduction of the likelihood ratio due to the allowance of multiple linkages implies that it is advantageous to form a highly localized target track. This is precisely what is done by the process described in conjunction with Fig. A-2. As a result the quantity (N) may be decreased as the track history builds up and the average number of samples required to make a decision when a target is indeed present may be decreased.

A.1.5 Approximations Used in the SLR Processor

As is so often the case, when it comes time to implement, approximations have to be employed. Some of the more important approximations that are used in the implementation of the SLR processor are presented in this section.

Consider first the likelihood ratio that was discussed earlier. This functional is defined by the probability density functions that hold at the sonar processor output for noise alone and signal-plus-noise. In the implementation that we have carried forth, it has been assumed that the sonar processor in question is a linear replica correlator followed by an envelope detector. In the presence of a Gaussian noise-plus-steady signal input the output probability density function is given by

$$p_1(x) = \frac{x}{\psi_n} \exp\left[-\frac{x^2 + s^2}{2\psi_n}\right] I_o(xs/\psi_o)$$

where

ψ_n = noise power at the correlator output

s = signal amplitude at the correlator output

I_o = modified Bessel function of zero order.

When noise alone is present, we have $s \equiv 0$ and the resulting probability density function becomes

$$p_0(x) = \frac{x}{\psi_n} e^{-\frac{x^2}{2\psi_n}} .$$

The likelihood ratio being the ratio of $p_1(x)$ to $p_0(x)$ becomes

$$L(x) = e^{-s^2/2\psi_n} I_0(sx/\psi_n);$$

if we assume that normalization has occurred (by AGC action or the equivalent) we can treat the envelope in terms of the quantity $x/\psi_n^{\frac{1}{2}} = y$. (This is precisely what ideal normalization does, i.e., it transforms x to $x/\psi_n^{\frac{1}{2}}$.) Thus, the likelihood ratio can be written,

$$L(y) = e^{-s^2/2\psi_n} I_0\left(y \frac{s}{\psi_n^{\frac{1}{2}}}\right).$$

If we define the peak signal-to-noise power ratio ρ at the correlator output as,

$$\rho^2 = \frac{s^2}{\psi_n},$$

the likelihood ratio becomes

$$L(y) = e^{-\frac{\rho^2}{2}} I_0(\rho y).$$

Using the asymptotic expansion of $I_0(z)$ for large values of z*--precisely the range of ρy where we will find that a linear approximation holds, we have for L(y) approximately

$$L(y) \approx e^{-\rho^2/2} \frac{e^{\rho y}}{(2\pi\rho y)^{\frac{1}{2}}}.$$

---

*M. Abramowitz and I. A. Stegun, Handbook of Mathematical Functions, Dover, 1965, p 377, Section 9.7.

Taking the natural logarithm of L(y) gives

$$\ln L(y) \approx \ell(y) \approx \rho y - (\rho^2/2 + \tfrac{1}{2} \ln 2\pi\rho y)$$

$$\approx \rho y - (\rho^2/2 + \tfrac{1}{2} \ln 2\pi + \tfrac{1}{2} \ln \rho y).$$

The property of the logarithm is such that for large values of ρy the log of ρy is much smaller than ρy, this last equation can be written as

$$\ell(y) \approx \rho y - (\rho^2/2 + \tfrac{1}{2} \ln 2\pi)$$

which is a linear approximation to the logarithm of the likelihood ratio. This function and the logarithm of the true likelihood ratio,

$$\ell(y) = \ln I_0(\rho y) - \rho^2/2$$

are plotted in Fig. A-3. In this figure it can be seen that over the range of y shown there is quite good agreement in the two functions. In practice we obtain even better agreement by adding an intercept correction factor to the linear approximation. Because of the excellent agreement in the slopes of the two curves this intercept correction suffices to bring the linear approximation into even better agreement.

The significant point here is that the parameter ρ^2, which is the assumed and hence the design signal-to-noise ratio, determines the slope and intercept of the linear approximation to the log likelihood ratio. That this dependence of the linear approximation on the design signal-to-noise ratio is an important relationship will be made clear in the following paragraphs.

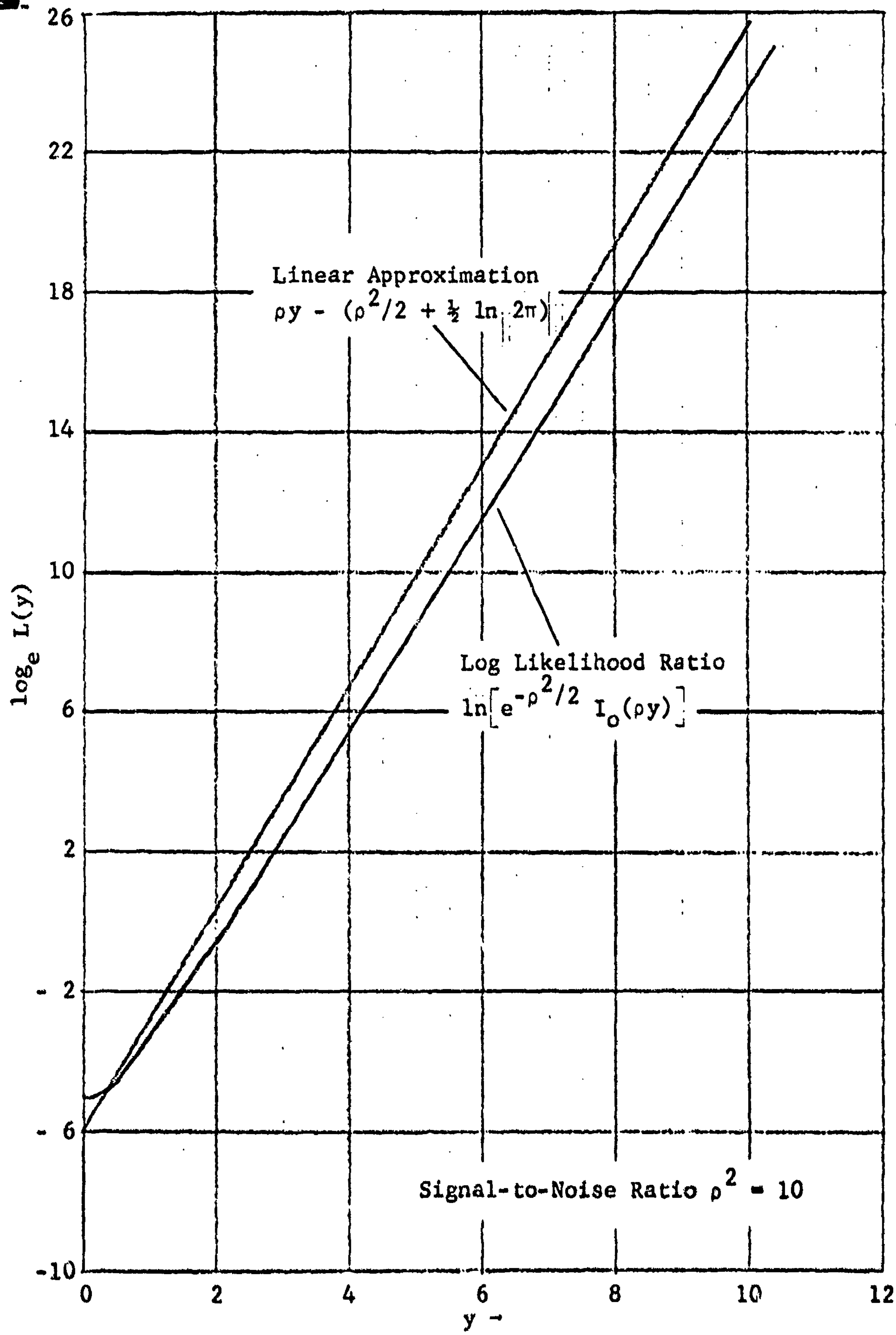


FIG. A-3 - A COMPARISON OF THE ACTUAL LOG LIKELIHOOD RATIO AND A LINEAR APPROXIMATION

Two other approximations that are used in the SLR processor implementation are the equations for the upper and lower thresholds, T_D' and T_L', respectively. Wald has shown that these thresholds are related to the error probabilities discussed earlier, α and β by the following relationships.

$$T_D' \leq \frac{1-\beta}{\alpha}$$

and

$$T_L' \geq \frac{\beta}{1-\alpha} .$$

Wald has also shown, however, that no appreciable increase in α and β occur when the following equalities are used,

$$T_D' = \frac{1-\beta}{\alpha}$$

and

$$T_L' = \frac{\beta}{1-\alpha} .$$

Quite simply this means that we suffer no appreciable change in detection and false alarm probabilities when these equalities are used. This is precisely how these thresholds are set in our implementation of the SLR processor. It should be noted, however, that since our processor operates on the basis of the logarithm of the likelihood ratio, the actual thresholds, T_D and T_L, that are used given by

$$T_D = \ln T_D'$$

and

$$T_L = \ln T_L'$$

Finally, we introduce another threshold that is used for practical purposes. This is the threshold that is applied to the sonar processor output data prior to entry into the computer SLR process. This thresholding is performed for the obvious reason that if all of the data emerging from the sonar output were entered into the computer, there would immediately arise a severe computer storage capacity problem. There are two aspects to this step that are of concern here. First, by thresholding the data there has clearly been a change in the distribution and density functions of the random process whose likelihood ratio we must know. Let us consider briefly how the likelihood ratio is affected by this thresholding process. Restating the likelihood ratio we have

$$L(x) = \frac{p_1(x)}{p_0(x)} .$$

Now, it will be shown that if data $\{x\}$ with continuous probability density function $h(x)$ is thresholded at an initial level x_I, then the resulting probability density function is given by,

$$f(x) = \begin{cases} \dfrac{h(x)}{1 - H(X_I)}, & x \geq X_I \\ 0 & x < X_I \end{cases}$$

where

$$H(X_I) = \int_{-\infty}^{X_I} h(x)\, dx .$$

In other words, the resulting probability density function is changed only by a scale factor, $1/(1 - H(X_I))$, for values of the random variable greater than or equal to the threshold.

The proof of this is quite simple and proceeds as follows. If $h(x)$ is a continuous probability density function and if a new random variable y is defined by

$$y = \begin{cases} x \text{ if } x \geq X_I \\ 0 \text{ otherwise} \end{cases}$$

then the probability that $y < Y$ and that $x \geq X_I$ is equal to

$$Pr[x \geq X_I,\ x < Y] = Pr[X_I \leq x < Y]$$

for

$$X_I < Y.$$

The conditional probability distribution of y given that $x \geq X_I$ is

$$F_y(Y \mid x \geq X_I) = \frac{Pr[X_I \leq x < Y_I]}{Pr[x \geq X_I]}$$

which from above reduces to

$$F_y(Y \mid x \geq X_I) = \frac{G_x(Y) - G_x(X_I)}{1 - G_x(X_I)} \quad .$$

Taking the derivative with respect to Y gives the probability density function y, thus

$$\begin{aligned} f_y(Y \mid x \geq X_I) &= \frac{g_x(Y)}{1 - G_x(X_I)}, & x \geq X_I \ . \\ &= 0 \qquad\qquad , & 0 < X_I \end{aligned}$$

Now referring back to the likelihood ratio definition we can write this likelihood ratio for thresholded data as,

$$L_T(x) = \frac{p_1(x)/(1 - F_1(X_I))}{p_o(X)/(1 - F_0(X_I))}$$

or

$$L_T(x) = L(x) \frac{1 - F_0(X_I)}{1 - F_1(X_I)} .$$

Calling the logarithm of $L_T(x)$, $\ell_T(x)$, we get

$$\ell_T(x) = \ell(x) + \ell n \frac{1 - F_0(X_I)}{1 - F_1(X_I)}$$

which shows that the likelihood ratio of the thresholded data is the same as that of the original data except for the additive constant, $\ln[(1 - F_0(X_I)/(1 - F_1(X_I))]$.

There remains the question of what value of X_I to use in thresholding the raw sonar output data. We have chosen to use a value of X_I which is given by the equation,

$$T_L = \rho X_I - \rho^2/2 - \text{constant}.$$

The reader will recognize that this is just the linear mapping that is used to approximate the log likelihood ratio. Our rationale for using this value of X_I is simply that a lower value would admit data to the SLR algorithm some of which would almost immediately be rejected since by definition, after the linear transformation it would be less than T_L. To choose a larger value would deny the

SLR algorithm some data which, when mapped, would exceed T_L and hence could possibly generate a legitimate track.

The second consequence of using an initial threshold value, X_I, that depends on the design signal-to-noise ratio is that as this latter quantity is varied so is X_I and so then is the amount of data that is entered into the computer. That is, the computer loading becomes a function of the design signal-to-noise ratio.

In the same way that computer loading depends on the design signal-to-noise ratio through X_I, the performance of the SLR is dependent on the design signal-to-noise ratio. This is true not only in the classical sense*, but in a special sense that derives from the search and tracking functions of the SLR algorithm. As discussed earlier, the algorithm uses projected search areas for finding data on one ping cycle that may be associated or linked with data on previous ping cycles in order to form a track. It is clear that if more noise samples are introduced into these search areas by lowering the initial threshold, then there will be more opportunities to generate spurious noise tracks. This represents a degradation in performance. It is also true that if the algorithm is modified to retain only a limited number of tracks by such a process as keeping only those N tracks which exhibit the largest log likelihood ratios**, then by lowering the initial threshold and thus passing more noise to the algorithm in a given search area the signal-to-noise ratio required to exceed the upper threshold, T_D, is increased. This

*John C. Hancock and Paul A. Wintz, Signal Detection Theory McGraw Hill Book Company, New York, 1966, pp 100-112.

**This is one of the modes of operation of the general SLR algorithm.

represents a degradation in detection performance. Fortunately, this is compensated for by the fact that a lower initial threshold also means that target echo plus noise is more likely to exceed this threshold. The point of all this is simply that the design signal-to-noise ratio is a rather critical parameter in the design of the SLR algorithm.

A.2 DESCRIPTION OF THE SLR COMPUTER PROCESS

A.2.1 Introduction

This section describes the computer process designed to accomplish SLR processing on multibeam output data from a sonar signal processor. This computer process is implemented on a UNIVAC 1108 digital computer. The characteristics of this particular implementation are such that the process may be implemented on a reasonably modest, state-of-the-art digital computer, such as can be found on board newer surface ships, or alternatively, on special purpose digital hardware.

The overall purpose of the SLR process is to produce a sonar display with reduced clutter, wherein the digital processor can perform ping-to-ping integration for any echo returns not large enough to display initially. In this manner, the sonar operator may remain alerted for longer periods of time, as well as becoming alerted earlier than with only the conventional processor. This process has been designed as a function which can be inserted into a conventional active sonar processing system between the output of the signal processor and the cathode ray tube (CRT) display. The primary requirement for its implementation is a digital computer with sufficient capacity, or specialized digital hardware.

The information flow in the SLR computer process is shown in Fig. A-4. The remainder of this section is devoted to a more detailed explanation of the process.

A.2.2 Preliminary Data Reduction

For the purpose of this explanation it is assumed that the output of the sonar signal processor is time and bearing

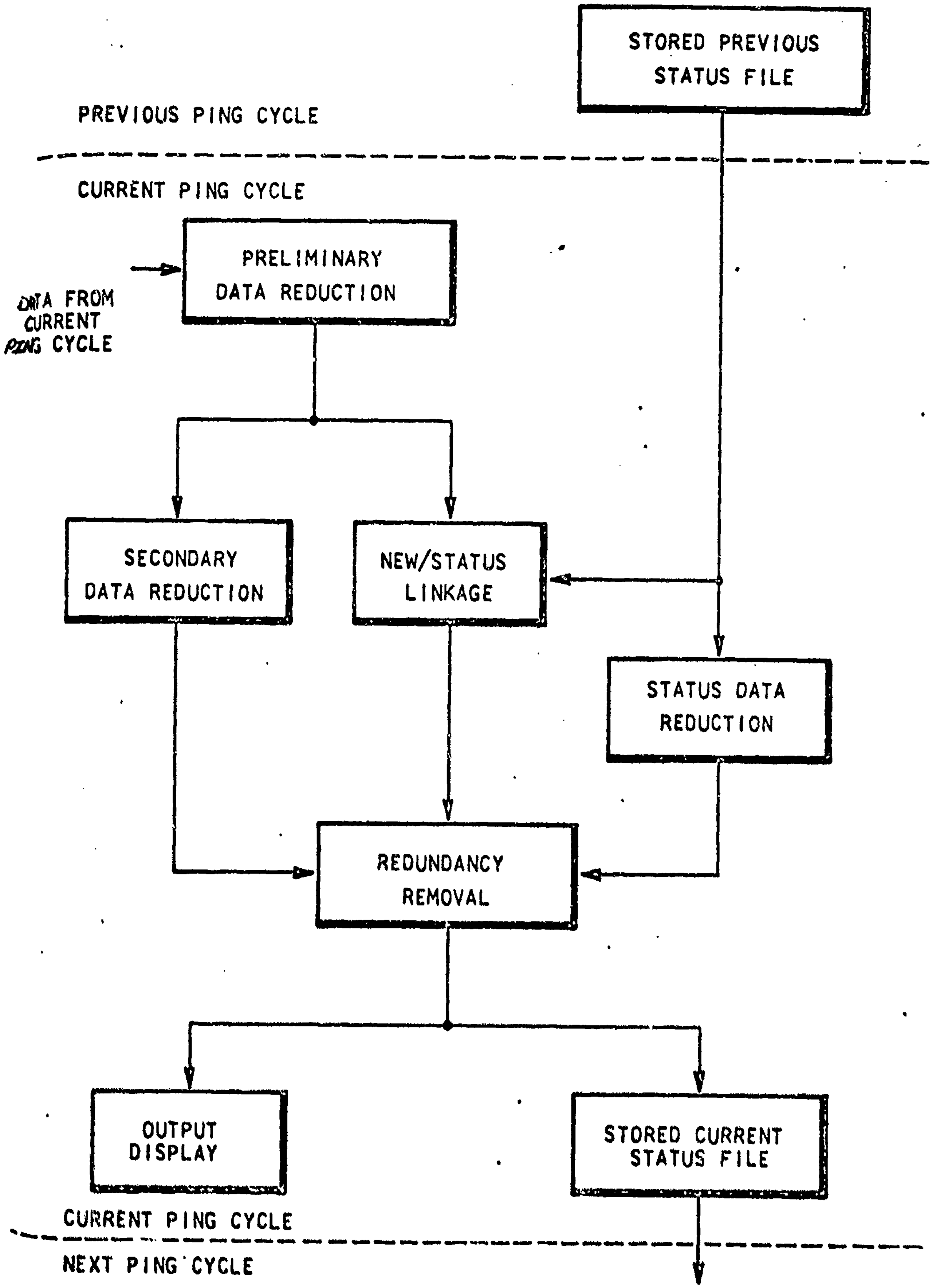


FIG. A-4 GENERAL DATA FLOW IN THE SLR PROCESS

DWG A6-129-12

normalized. Thus, the normalized data from the current ping cycle are processed first by the Preliminary Data Reduction section. This section has three purposes. First, the data received are grouped into single-ping event packages and, if necessary, are converted from analog to digital format. These single ping event packages contain information such as the range, bearing and amplitude of each data point from the processor output. In order to facilitate digital computer processing with the SLR method, the parameters which describe a single ping event package are divided into resolution cells, each corresponding to an incremental range of the dimension of interest. For example, a mark on the display may represent 50 yards, hence a suitable definition of a range resolution cell. These resolution cells may be adjusted to comply with both the sonar system and the computer available.

Second, the section performs a preliminary or initial thresholding function mentioned earlier. Third, the amplitude of each data point which passed the preliminary threshold is mapped to the logarithm of its likelihood ratio, using a linear approximation discussed previously. The output of the Preliminary Data Reduction section is passed to two sections, New/Status Linkage, and Secondary Data Reduction.

A.2.3 New/Status Linkage

The New/Status Linkage section receives two inputs; one is the reduced single ping sonar output from the Preliminary Data Reduction section, and the other is the series of multiping event packages containing the joint log likelihood ratio, range-bearing coordinates, and projected search area from the previous ping status file. Each event package is called one status unit.

A.2.3.1 Status File - Each event package, or status unit stored in the status file is represented by four functional quantities which are listed below:

1. The event position vector are coordinates from the preceding echo cycle;

2. The expected event position vector for the current echo cycle;

3. The search area or volume for the current echo cycle; and

4. The joint log likelihood ratio resulting from the previous echo cycles.

The number of dimensions of position vectors and search areas depends upon the sonar system. That is, the number of dimensions depends upon the number of quantities that can be measured for each sample by the sonar system. The search area or volume defines the region centered about the expected position vector within which legitimate linkages can occur during the current echo cycle with the event logged in the status unit.

A.2.3.2 Linkage Process - The New/Status Linkage section compares each status unit with the single ping event packages from the reduced sonar output. If the single ping event position vector lies within the search area of the status unit, the single ping event is said to be linked with the status file entry. When this situation occurs, the joint log likelihood ratio of the new multiping event is formed by the process described earlier in this appendix and is then tested against the lower decision threshold, T_L.

If this new joint log likelihood ratio is greater than T_L, then a new status unit is formed, with information from

the old status unit being processed in conjunction with the single ping event package to generate a new event position vector, a new estimated position vector, and a new search area for the new status unit. If the new joint log likelihood ratio is less than T_L, then hypothesis H_0 is chosen and the track linkage is discarded, precluding the calculation of a new status unit.

A status unit is allowed to link with all events which fall within its projected search area. Similarly, a single ping event can fall within the search areas of several status units and hence be linked several ways. This procedure allows many incorrect linkages, but since all incorrect linkages will yield a noise track, the process will decrease the log likelihood ratio and the track will eventually be dropped. The process will reach a steady-state condition in which as many noise tracks are being discarded as are being added, on the average.

A.2.4 Secondary Data Reduction

The reduced sonar output from the Preliminary Data Reduction section is also processed by the Secondary Data Reduction section. The Secondary Data Reduction section tests the log likelihood ratio of the single ping event package against the lower decision threshold, T_L, and makes the appropriate decision. If indeed the single ping event exceeds the threshold, a new status unit is created on a single ping basis, except that the search area is larger than for most multiping status units, since there is not as much information regarding an expected position vector in the sense of a multiping status unit.

Upon initialization of the SLR computer process, there are no previously acquired status units, hence the Secondary Data Reduction section is the only section capable of producing a status unit. In each echo cycle, it is here that new tracks are

started. Note that the entire process does not prevent a single large echo from being entered into the status file and being placed upon the output display immediately.

A.2.5 Status Data Reduction

The status file information is utilized in two ways in the SLR computer process. As described above, each status unit is furnished to the New/Status Linkage section to determine linkages and form target tracks. Also, the entire status file is passed through the Status Data Reduction section. The purpose of this section is to maintain a strong target track even though the current echo cycle did not produce a linkage with this track.

This function is accomplished by assuming that each status unit linked with a small single ping event whose log likelihood ratio was just below T_L, and whose position vector was the same as the expected position vector of the status unit being processed. The search area is enlarged to accommodate the increased uncertainty of target position, and a possible new status unit is formed. The log likelihood ratio of the new status unit is tested against T_L, and the appropriate decision is made. If the new status unit exceeds the threshold, it is passed to the next processing section. This procedure helps to avoid losing a well-established track because of a single miss, yet a noise track is discarded quickly because of the degradation.

A.2.6 Redundancy Removal

From the above discussion, it can be seen that there are three sections in the SLR process capable of producing status units to be entered into the current status file. The three sections are listed below:

1. New/Status Linkage;
2. Secondary Data Reduction; and
3. Status Data Reduction

Since these three sections operate independently in generating possible status units, there is a possibility that some of the status units will be redundant, that is, several may have the same predicted location vector and the same present location vector, in terms of resolution cells. This redundancy can be caused in a number of ways. For example, a single ping entry may be formed, a linkage also formed with the single ping entry and a track propagation entry may be formed, all with the same present and expected position vectors. The redundancy removal section scans all entries to determine these redundancies and removes all except the status unit with the largest log likelihood ratio.

The output of the Redundancy Removal section is the new status file for the current echo cycle. This is placed in storage for the next echo cycle, and is made available to the Output Display.

A.2.7 Output Display

The Output Display section is assumed to be part of the original sonar system. Hence, the operator should have control of the display threshold, T_D. By increasing this threshold, the operator can reduce the clutter to a more acceptable rate with the SLR and retain the same information as without the SLR processor. When the operator becomes alerted, he can lower the display threshold in order to look at the status file in more detail, since a change in the display threshold immediately changes what information is displayed. There is no need to wait

for past events to accumulate on the display, since the accumulation has already occurred and is stored in the status file.

Note that the SLR processor does not include a fundamental specification of the number of echo cycles over which integration will be carried. Rather, a single status unit could represent a track that has been carried for an indefinite number of pings. Note also that a change in the lower decision threshold does not affect the degree of clutter on the display, but only the amount of processing and storage. Hence there is significant improvement over conventional approaches which allow ping-to-ping integration only through the operator looking at the display, in which it is necessary to operate with a clutter rate sufficient to allow small echoes to mark the display so that the ping-to-ping integration process may begin.

TRACOR 6500 TRACOR LANE, AUSTIN, TEXAS 78721

APPENDIX B

A SLR PROCESSOR FOR ARL'S CDC 3200 COMPUTER

B1.0 INTRODUCTION

A set of operating instructions together with the complete program listings and flow diagrams are presented for the active SLR program that was implemented on ARL's CDC 3200 computer.

B2.0 OPERATING INSTRUCTIONS

Input data required to run the SLR program are obtained from magnetic tape and from data cards. The tape contains the detector outputs from the ARL digital sonar system. The tape consists of one 121 word identification record followed by one data record for each ping cycle. The quantities contained in each of the records are defined in the proper order as follows:

Identification Record for Detector Output

Word	Description
1	Length of Data Record
2	Sequence Number
4	Number of Data Records
5	Cursor Bearing (degree)
8	Analog Tape Number
9	Initial Analog Footage
10	Day-Month-Year
13	Ping number Relative to Initial Footage
14	Threshold
19	Data Function (4000_8 → detector output)
20	Data Tape or Source (600_8 → beamformer)
22	Time of Day (hour-minute-second)
29	Elapsed Time Since Previous Ping
30	Data Ship
31-60	Ship's Speed (nearest knot)
61-90	Ship's Course (degree, true)
91	Run Number
92	Expected Target Bearing (degree)
93	Expected Target Range (yards)
119-120	Program Name
121	63473164_8

Data Record For Detector Output

Word	Description
1	Range for First Peak (yards)
2	Bearing for First Peak (nearest degree)
3	Amplitude for First Peak
4	Sum of Sample Values in Normalizing Annulus (greater than one beam width from first peak)
5	Sum of sample values in normalizing annulus (within one beam width of first peak)
6	Largest Adjacent Sample (on same beam as first peak)

The above 6 words are repeated for every peak, making a total of 6N words per data record.

Card inputs to the SLR program define parameters needed to calculate the likelihood ratios as well as control parameters that define various program options. These are listed below:

Card	Columns	Description
1 (6I10)	1-10	IAA: Scaled slope of log likelihood ratio
	11-20	IBB: Scaled intercept of log likelihood ratio
	21-30	ITHRES: Scaled lower decision threshold
	31-40	ITBR: Scaled upper threshold
	41-50	LIKMAX: Maximum allowable log likelihood ratio

	51-60	TDB: Input data threshold (dB)
	61-70	SCLE: Scale factor for log likelihood ratio
2 (5I10)	1-10	IPRNT: 1-Print status file 0-No print
	11-20	ITOUT: 1-Output to tape 0-No tape output
	21-30	LUN: Logical unit number for tape output
	31-40	ISEQ: Positions output tape to begin next record
	41-50	ITIN: 1-Read initial status file from sequence number ISEQ on output tape (restarts a run) 0-Do not read initial status file
3 (6I5)	1-5	NVR(1): Range width for small tracking window
	6-10	NVB(1): Bearing width for small tracking window
	11-15	NVR(4): Range width for large tracking window
	16-20	NVB(4): Bearing width for large tracking window
	21-25	NVB(4): Number of independent samples for small window
	26-30	INDL: Number of independent samples for large window
4 (I5)	1-5	Input tape number

5 (3I5)	1-5	Input tape number
	6-10	First sequence number to be processed
	11-15	Last sequence number to be processed.

Cards 4 and 5 are read in through system subroutines SEQCARD and SEQIN defined by ARL's CDC 3200 software package.

The SLR program listing and flow diagrams were generated by a TRACOR UNIVAC 1108 special purpose utility program and are presented as follows:

```
      C      PROGRAM SLR IS THE MAIN PROGRAM FOR EXECUTING SEQUENTIAL LIKELI-
      C      HOOD RATIO PROCESSING ON ARL'S CDC 3200 COMPUTER.   IT OPERATES
      C      ON THE OUTPUT OF ARL'S DIGITAL SONAR, USING THE DIMENSIONS OF
      C      RANGE(YDS) AND BEARING(DEG) TO TRACK POSSIBLE TARGETS.  THE DATA
      C      INPUT IS FROM MAGNETIC TAPE.  DISK STORAGE IS USED AS NECESSARY
      C      TO GIVE ADDITIONAL SPACE FOR TRACK INFORMATION STORAGE.
      C      ISU CONTAINS TRACK INFOFMATION IN THE FOLLOWING SEQUENCE.  PRESENT
      C      RANGE, PRESENT BEARING, PREDICTED RANGE, PREDICTED BEARING, VARI-
      C      ANCE INDICATOR, SCALED JOINT LOG LIKELIHOOD RATIO.
      C      NVR AND NVB CONTAIN THE TRACKING WINDOW SIZE FOR RANGE AND BEARING
      C      IVAR CONTAINS THE LOG LIKELIHOOD RATIO CORRECTION DUE TO TRACKING
      C      WINDOW SIZES.
      C      JS AND IS STORE THE ARRAY LOCATION OF LARGE LOG LIKELIHOOD RATIO
      C      TRACKS FOR SPECIAL PROCESSING.
             COMMON  ISU(10104),IBUFF(12000),ID(122)
      C      IBUFF IS CORE AREA FOR INPUT DATA.
      C      ID CONTAINS TAPE AND PING INFORMATION.
             DIMENSION NVR(5),NVB(5),IVAR(5),JS(20),IS(20)
      C      INITIALIZE STORAGE
             CALL EQDISK(5,32000)
             RAD=.0174533
             NW=6
             KKMAX=5052-NW
             JJMAX=60096-NW
             JSTART=0
             B=JJMAX*.05
             A=.9*JJMAX/20000
             KDR=200
      C      READ INPUT DATA
         160 READ  900,IAA,IBB,ITHRES,ITBR,LIKMAX,TDB,SCLE
      C      IAA,IBB - SCALED SLOPE AND INTERCEPT OF LOGLIKELIHOOD RATIO.
      C      ITHRES - SCALED LOWER DECISION THRESHOLD.
      C      ITBR - SCALED UPPER THRESHOLD ABOVE WHICH TRACKS ARE SUBJECTED
      C             TO SPECIAL PROCESSING.
```

```
C     LIKMAX - MAXIMUM ALLOWABLE LOG LIKELIHOOD RATIO.
C     TDB - INPUT DATA THRESHOLD IN DB.
C     SCLE - SCALE FACTOR FOR LOG LIKELIHOOD RATIO.
C     OUTPUT OPTIONS  1 - SELECT, 0 - NOT SELECT
C     IPRNT - PRINT STATUS FILE
C     ITOUT - OUTPUT TO MAGNETIC TAPE
C     LUN - LOGICAL UNIT NUMBER FOR TAPE OUTPUT.
      IF(IAA.LE.0)  GO TO 400
      READ 900,IPRNT,ITOUT,LUN,ISEQ,ITIN
      THRESA=10.**(TDB/20.)
      SHRESA=THRESA
C     INITIALIZE TRACKING WINDOWS
      READ 901,NVR(1),NVB(1),NVR(4),NVB(4),INDS,INDL
      NT=INDS
      IVAR(1)=ALOG(FLOAT(NT))*SCLE
      DO 150 I=2,3
      NVR(I)=2*NVR(I-1)
      NVB(I)=2*NVB(I-1)
      NT=2*NT
  150 IVAR(I)=ALOG(FLOAT(NT))*SCLE
      IVAR(4)=ALOG(FLOAT(INDL))*SCLE
      IVAR(5)=ALOG(FLOAT(INDL*2))*SCLE
      NVR(5)=2*NVR(4)
      NVB(5)=2*NVB(4)
      NVRH=NVR(1)/2
      NVBH=NVB(1)/2
  900 FORMAT(5I10,3F10.4)
  901 FORMAT(16I5)
      KOLD=1
      JOLD=1+JSTART
      JNEW=JJMAX+JSTART+NW+1
      KNEW=KKMAX+NW+1
      KNT=KNEW
      JNT=JNEW
      XG=0.
      YG=0.
      IF(ITOUT.LT.1.AND.ITIN.LT.1.OR.ISEQ.LT.1) GO TO 144
      CALL FINDSEQ(LUN,ISTAT,ISEQ,ID)
      IF(ISTAT.GT.0) GO TO 30?
      NT=1
```

```
146 CALL INPUTAPE(LUN,ISTAT,KKMAX+NW,1,0,NT,NT+ID(1)-1,1,ISU(KNEW))
    NT=NT+ID(1)
    IF(ISTAT.GT.0) GO TO 300
    IF(ID(1).GE.ID(118)) GO TO 145
    CALL DRUM(ID(1),ISU(KNEW),JNT,1)
    IF(JNT-JNEW.LT.ID(118)) GO TO 146
    JNT=JNEW+ID(118)
145 KNT=KNT+ID(1)
    IF(ITIN.GT.0) GO TO 144
    KNT=KNEW
    JNT=JNEW
144 CONTINUE
    PRINT 906,IAA,IBB,ITHRES,ITBR,LIKMAX,TDB,SCLE,THRESA
906 FORMAT(22H SLP PROCESSING SLOPE=,I10,11H INTERCEPT=,I10,/
   111H LIK THRES=,I10,13H TRACK THRES=I10,10H MAX LIK=,I10,/,
   25H TDB=F10.4,6H SCLE=,F10.4,8H THRESA=,F10.4)
    PRINT 907
907 FORMAT(7X,3HNVR,7X,3HNVB,6X,4HIVAR)
    PRINT 902,(NVR(I),NVB(I),IVAR(I),I=1,5)
902 FORMAT(3I10)
    PRINT 902,NVRH,NVBH
909 FORMAT(27H INPUT/OUTPUT OPTIONS PRNT=I2,6H TOUT=,I2,5H LUN=,I2,
   16H ISEQ=,I4,5H TIN=,I2)
    PRINT 909,IPRNT,ITOUT,LUN,ISEQ,ITIN
C   INITIALIZE TAPE ROUTINE AND COUNTERS FOR RUN
    CALL SEQCARD(ITAPE,NPING,NPMAX,ITHSEQ,IROM,ITO,ITH,ITYPE,NCHAN,
   1ISEARCH,ID,IEOF)
    IF(IEOF) 160,140,300
140 PRINT 908,ITAPE,NPING,ITHSEQ,NPMAX,IROM,ITH,ITO,ITYPE,NCHAN,
   1ISEARCH,ID(1),ID(2)
908 FORMAT(21H INPUT TAPE.  NUMBER=I5,10H SEQUENCE ,I5,1H(,I5,1H),I5,
   1/7H WORDS ,I7,1H(,I5,1H),I7,6H TYPE=I3,7H NCHAN=I5,8H SEARCH=I3,/
   24H ID ,2I10)
C   INITIALIZE COUNTERS AND STORAGE FOR PING CYCLE.
100 NT=KNEW
    KNEW=KOLD
    KOLD=NT
    NT=JNEW
    JNEW=JOLD
    JOLD=NT
```

```
      KMAX=KNEW+KKMAX
      JMAX=JNEW+JJMAX
      JHP=JNT
      JNT=JNEW
      JCNT=JOLD
      IF(JHP.LE.JOLD) GO TO 110
      I=KKMAX+NW
      IF(JHP-JOLD.LT.I) I=JHP-JOLD
      CALL DRUM(I,ISU(KOLD),JCNT,0)
      KHP=KOLD+I-NW
      GO TO 115
  110 KHP=KNT-NW
  115 LR=0
      LB=0
      LU=KOLD
      LL=KOLD
      KSS=KNEW
      KNT=KNEW
      KR=KDR
      JSM=0
      JS(1)=0
      IREC=1
    3 IROM=(IREC-1)*IROM+1
      ITO=IROM+11999
      ICNT=-5
C     READ INPUT DATA FOR PING CYCLE
      CALL SEQIN(ITAPE,NPING,ISTAT,IBUFF)
      IF(IREC.GT.1) GO TO 7
      X=ID(61)*RAD
      T=ID(29)*ID(31)*.00056258
      XG=XG+T*SIN(X)
      YG=YG+T*COS(X)
      X=ID(92)*RAD
      T=XG+ID(93)*SIN(X)
      X=YG+ID(93)*COS(X)
      ID(95)=SGRT(X*X+T*T)
      ID(94)=ATAN(T/X)/RAD
      IF(X.GE.0) GO TO 5
      ID(94)=ID(94)+180
      GO TO 6
```

```
    5 IF(T.GE.0) GO TO 6
      ID(94)=ID(94)+360
    6 PRINT 910,ID(1),ID(2),ID(4),ID(61),ID(31),(ID(I),I=91,95)
  910 FORMAT(16H INPUT DATA WDS=I6,5H SEQ=I4,10H DATA REC=I3,6H CRSE=I4,
     15H SPD=I3,5H RUN=I3,9H R/R REL=I4,I6,5H GEO=I4,I6)
      ID(96)=XG
      ID(97)=YG
      NSRL=ID(95)-500
      NSRH=ID(95)+500
      NSBL=ID(94)-15
      NSBH=ID(94)+15
    7 IF(ISTAT) 8,8,300
    8 ICNT=ICNT+6
      IF(ICNT.LE.ID(1)) GO TO 2
      IF(IREC.GE.ID(4)) GO TO 200
C     FIND SAMPLE LOCATION AND LOG LIKELIHOOD RATIO
      IREC=IREC+1
      GO TO 3
    2 IR=IBUFF(ICNT)
      IB=IBUFF(ICNT+1)
      T=((IBUFF(ICNT+2)+IBUFF(ICNT+5))/2+IBUFF(ICNT+3)/42.-IBUFF(ICNT+4)
     1/22.)/((IBUFF(ICNT+3)+IBUFF(ICNT+4))/64.)
      IF(T.LT.THRESA) GO TO 8
      LIK=IAA*T+IBB
    4 IF(IR.LT.KR) GO TO 9
C     ADAPTIVE INPUT THRESHOLD
      DIF=FLOAT(JNT-JNEW+KNT-KNEW)/(KR*A+B)-1.
      THRESA=THRESA*(1.+DIF*ABS(DIF))
      IF(THRESA.LT.SHRESA) THRESA=SHRESA
      KR=KR+KDR
      GO TO 4
    9 IF(KHP.LE.KOLD) GO TO 50
C     FIND LIMITS ON STORED TRACK INFORMATION AND UPDATE LARGE TRACK
C     LOCATIONS.
      NT=LR-NVR(4)
   10 IF( ISU(LL).GT.NT) GO TO 20
      IF(LL.NE.JS(1)) GO TO 15
      JSM=JSM-1
      DO 11 K=1,JSM
      JS(K)=JS(K+1)
```

```
   11 IS(K)=IS(K+1)
      JS(JSM+1)=0
      IF(JSM.LT.19) GO TO 15
      INW=0
      K=1
      DO 12 J=LL,LU,NW
      IF(JS(K).NE.J) GO TO 13
      K=K+1
      GO TO  12
   13 IF(ISU(J+5).LE.INW) GO TO 12
      INW=ISU(J+5)
      I=J
      M=K
   12 CONTINUE
      IF(INW.LT.ITBR) GO TO 15
      JSM=JSM+1
      K=JSM
   14 K=K-1
      JS(K+1)=JS(K)
      IS(K+1)=IS(K)
      IF(K.GT.M) GO TO 14
      JS(M)=I
      IS(M)=0
   15 LL=LL+NW
      IF(LL.LT.KHP) GO TO 10
      LL=KHP
   20 NT=IR+NVR(4)
   21 IF(ISU(LU).GT.NT)  GO TO 30
      IF(ISU(LU+5).LT.ITBR.OR.LU.EQ.KHP) GO TO 25
      IF(JSM.LT.20) GO TO 24
      I=JS(1)
      M=1
      INW=ISU(I+5)
      DO 22 K=2,JSM
      I=JS(K)
      IF(ISU(I+5).GE.INW) GO TO 22
      INW=ISU(I+5)
      M=K
   22 CONTINUE
      IF(ISU(M+5).GE.ISU(LU+5)) GO TO 25
```

```
      JSM=JSM-1
      DO 23 K=M,JSM
      JS(K)=JS(K+1)
   23 IS(K)=IS(K+1)
   24 JSM=JSM+1
      JS(JSM)=LU
      IS(JSM)=0
   25 LU=LU+NW
      IF(LU.LT.KHP) GO TO 21
      LU=KHP
   30 K=1
      DO 40 J=LL,LU,NW
      L=0
      IF(J.NE.JS(K)) GO TO 31
      L=K
      IF(K.LT.JSM) K=K+1
   31 ISS=ISU(J+4)
C     PROPAGATE TRACKS AS NECESSARY
      IF(IORD(ISU(J+2),ISU(J+3),LR,LB ).LT.0.OR.IORD(ISU(J+2),ISU(J+3),
     1IR,IB).GT.0) GO TO 35
      NT=ISU(J+5)+ITHRES-IVAR(ISS)
      IF(NT.LT.ITHRES) GO TO 35
      IF(ISS.EQ.3.OR.ISS.EQ.5) GO TO 35
      IF(KNT    .GT.KMAX) GO TO 59
      CALL SU(ISU(KNT),ISU(J+2),ISU(J+3),ISU(J),ISU(J+1),ISU(J+4)+1,NT)
      KNT=KNT+NW
C     CHECK FOR TRACK LINKAGE
   35 IF(ICOMP(IR,IB,ISU(J+2)-NVR(ISS),ISU(J+3)-NVB(ISS),ISU(J+2)+
     1NVR(ISS),ISU(J+3)+NVB(ISS)).NE.0) GO TO 40
C     CALCULATE NEW LOG LIKELIHOOD RATIO
      NT=ISU(J+5)+LIK-IVAR(ISS)
      IF(NT.LT.ITHRES) GO TO 40
      IF(L.EQ.0) GO TO 39
      IF(IS(L).EQ.0) GO TO 38
   36 I=IS(L)
      IF(NT.LT.ISU(I+5)) GO TO 40
      ISU(I+5)=-99999
   38 IF(KNT.LE.KMAX) IS(L)=KNT
   39 IF(KNT.GT.KMAX) GO TO 59
      IF(ISU(J+5).GT.LIKMAX) NT=LIKMAX-IVAR(ISS)+LIK
```

```
C     CREATE NEW MULTIPING TRACK PACKET
      CALL SU(ISU(KNT),IR,IB,ISU(J),ISU(J+1),1,NT)
      KNT=KNT+NW
   40 CONTINUE
      LR=IR
      LB=IB
   50 IF(LIK.LT.ITHRES) GO TO 59
      IF(KNT.GT.KMAX) GO TO 59
C     CREATE SINGLE PING TRACK PACKET
      CALL SU(ISU(KNT),IR,IB,IR,IB,4,LIK)
      KNT=KNT+NW
C     TRACK REDUNDACY REMOVAL AND ORDERING SECTION.
   59 IKNT=KNT
   60 IKNT=IKNT-NW
      IF(IKNT.LT.KSS) GO TO 80
   61 KNTR=IKNT
   62 KNTR=KNTR-NW
      IF(KNTR.LT.KNEW) GO TO 60
      NT=ICOMP(ISU(KNTR),ISU(KNTR+1),ISU(IKNT)-NVRH,ISU(IKNT+1)-NVBH,
     1ISU(IKNT)+NVRH,ISU(IKNT+1)+NVBH)
      IF(NT.LT.-1.AND.KNTR.LE.KSS) GO TO 60
      IF(NT.GT.1) GO TO 70
      IF(NT) 62,63,73
   63 IF(ISU(IKNT+5).GT.ITBR.OR.ISU(KNTR+5).GT.ITBR) GO TO 64
      NT=ICOMP(ISU(KNTR+2),ISU(KNTR+3),ISU(IKNT+2)-NVRH,ISU(IKNT+3)-NVBH
     1,ISU(IKNT+2)+NVRH,ISU(IKNT+3)+NVBH)
      IF(NT.EQ.0)  GO TO 64
   73 IF(IORD(ISU(KNTR),ISU(KNTR+1),ISU(IKNT),ISU(IKNT+1))) 62,62,70
   64 IF(ISU(KNTR+4).EQ.4.OR.ISU(IKNT+4).EQ.4) GO TO 73
      KNT=KNT-NW
      IF(ISU(KNTR+5).GT.ISU(IKNT+5)) GO TO 69
      JL=KNTR
      IF(JL.LT.KSS) KSS=KSS-NW
   65 NT=KNT-1
      DO 66 I=JL,NT
      INW=I+NW
   66 ISU(I)=ISU(INW)
      GO TO 67
   69 JL=IKNT
      IF(KNT.NE.IKNT) GO TO 65
```

```
67 IF(JSM.EQ.0) GO TO 60
   DO 66 K=1,JSM
   IF(IS(K)-JL) 68,71,72
71 IS(K)=0
   GO TO 68
72 IS(K)=IS(K)-NW
68 CONTINUE
   GO TO 60
70 DO 75 I=1,NW
   INW=KNTR+I-1
   NT=ISU(INW)
   M=IKNT+I-1
   ISU(INW)=ISU(M)
75 ISU(M)=NT
   IF(JSM.EQ.0) GO TO 61
   I=0
   M=0
   DO 76 K=1,JSM
   IF(IKNT.EQ.IS(K)) I=K
   IF(KNTR.EQ. IS(K)) M=K
76 CONTINUE
   IF(I.NE.0) IS(I)=KNTR
   IF(M.NE.0) IS(M)=IKNT
   GO TO 61
80 KSS=KNT
C     DISK INPUT/OUTPUT OF TRACK INFORMATION
   IF(KNT+300.LT.KMAX) GO TO 191
99 NT=KNT-NW
   I=NT
98 I=I-NW
   IF(ISU(I).LT.ISU( NT)-NVRH) GO TO 91
   IF(I.GT.KNEW) GO TO 98
91 IF(JSM.EQ.0) GO TO 95
   M=0
   DO 92 K=1,JSM
   IF(I.LT.IS(K)) GO TO 92
   I=IS(K)-NW
   M=K
92 CONTINUE
   IF(I.GT.KNEW) GO TO 95
```

```
      I=KNEW
      IF(M.NE.0) IS(M)=0
   95 I=I-KNEW+NW
      IF(I+JNT.GE.JMAX) I=JJMAX+NW-JNT,JNEW
      CALL DRUM(I,ISU(KNEW),JNT,1)
      NT=KNEW+I-1
      IF(IPRNT.EQ.1) PRINT 903,(ISU(K),K=KNEW,NT)
  903 FORMAT(1X,4(I5,I4,I6,I4,I1,I6,1H,))
      DO 122 K=KNEW,NT,NW
      IF(ICOMP(ISU(K),ISU(K+1),NSRL,NSBL,NSRH,NSBH).NE.0) GO TO 122
      INW=K+NW-1
      PRINT 903,(ISU(L),L=K,INW)
  122 CONTINUE
      IF(JNT.EQ.JMAX) GO TO 300
      KNT=KNT-I
      KSS=KSS-I
      NT=KNT-1
      DO 93 J=KNEW,NT
      INW=J+I
   93 ISU(J)=ISU(INW)
      DO 190 K=1,JSM
      IF(IS(K).NE.0)  IS(K)=IS(K)-I
  190 CONTINUE
  191 IF(LU.LT.KHP) GO TO 8
      IF(JCNT.GE.JHP) GO TO 8
      IF(LL.EQ.KOLD) LL=KOLD+NW
      I= LL-KOLD
      NT=KHP-I+NW-1
      DO 192 K=KOLD,NT
      INW=K+I
  192 ISU(K)=ISU(INW)
      LL=LL-I
      LU=LU-I
      IF(JSM.EQ.0)  GO TO 193
      IF(JS(1)-I.GE.LL) GO TO 194
      JSM=JSM-1
      DO 196 K=1,JSM
      IS(K)=IS(K+1)
  196 JS(K)=JS(K+1)-I
      IS(JSM+1)=0
```

```
      JS(JSN+1)=0
      GO TO 193
  194 DO 195 K=1,JSN
  195 JS(K)=JS(K)-I
  193 NT=NT+1
      IF(I.GT.JHP-JCNT+NW) I=JHP-JCNT
      KHP=NT+I-NW
      CALL DRUM(I,ISU(NT),JCNT,0)
      GO TO 8
C     END OF PING CYCLE CLEAN-UP
  200 IF(JNT.EQ.JNEW) GO TO 220
      I=KNT-KNEW
      IF(I+JNT.GT. JMAX) I=JMAX+NW-JNT
      CALL DRUM(I,ISU(KNEW),JNT,1)
  220 MT=(KNT-KNEW)/NW
      NT=(JNT-JNEW)/NW
      PRINT 911,NPING,MT,NT
  911 FORMAT(10H PING NO.=,I4,19H NO. OF SU IN CORE=,I6,9H ON DISK=,I6)
      I=KNT-1
      IF(IPRNT.EQ.1) PRINT 903,(ISU(K),K=KNEW,I)
      DO 222 K=KNEW, I,NW
      IF(ICOMP(ISU(K),ISU(K+1),NSRL,NSRL,NSRH,NSBH).NE.0) GO TO 222
      INW=K+NW-1
      PRINT 903,(ISU(L),L=K,INW)
  222 CONTINUE
      IF(ITOUT) 240,240,225
  225 IF(JNT.EQ.JNEW) GO TO 230
      ID(1)=KKMAX+NW
      ID(118) =JNT-JNEW
      IF(ID(1).GT.ID(118)) ID(1)=ID(118)
      IF(ID(4)*ID(1).LT.ID(118)) ID(4)=ID(4)+1
      NT=ID(118)/NW
      ID(1)=NT/ID(4)
      IF(ID(4)*ID(1).LT.NT) ID(1)=ID(1)+1
      ID(1)=ID(1)*NW
      JCNT=JNEW
      CALL DRUM(ID(1),ISU(KNEW),JCNT,0)
      GO TO 235
  230 ID(1)=KNT-KNEW
      IF(ID(1).GT.0) GO TO 232
```

```
      ID(1)=NW
      DO 231 I=1,NW
      J=KNEW+I-1
  231 ISU(J)=0
  232 ID(4)=1
      ID(118)=ID(1)
      JCNT=JNEW+ID(1)
  235 CALL TAPEOUT(LUN,ISTAT,ID(1),ISU(KNEW))
      IF(ISTAT.GT.0) GO TO 300
  236 IF(JCNT.GE.JNT) GO TO 240
      CALL DRUM(ID(1),ISU(KNEW),JCNT,0)
      CALL DATAOUT(LUN,ISTAT,ISU(KNEW))
      IF(ISTAT) 236,236,300
  240 NPING =NPING+ITHSEG
      IF(NPING-NPMAX) 100,100,160
  300 PRINT 905,IEOF,ISTAT
  905 FORMAT(18H INPUT ERROR IEOF=,I5,7H ISTAT=,I5)
  400 CONTINUE
      END
```

0 DIAGNOSTIC MESSAGES.

1

```
---------------------------------------
   PROGRAM SLR IS THE MAIN
   PROGRAM FOR EXECUTING
   SEQUENTIAL LIKELI-

   HOOD RATIO PROCESSING ON ARL'S
   CDC 3200 COMPUTER.   IT
   OPERATES

   ON THE OUTPUT OF ARL'S DIGITAL
   SONAR, USING THE DIMENSIONS OF

   RANGE(YDS) AND BEARING(DEG) TO
   TRACK POSSIBLE TARGETS.  THE
   DATA

   INPUT IS FROM MAGNETIC TAPE.
   DISK STORAGE IS USED AS
   NECESSARY

   TO GIVE ADDITIONAL SPACE FOR
   TRACK INFORMATION STORAGE.

   ISU CONTAINS TRACK INFORMATION
   IN THE FOLLOWING SEQUENCE.
   PRESENT

   RANGE, PRESENT BEARING,
   PREDICTED RANGE, PREDICTED
   BEARING, VARI-

   ANCE INDICATOR, SCALED JOINT
   LOG LIKELIHOOD RATIO.

   NVR AND NVB CONTAIN THE
   TRACKING WINDOW SIZE FOR RANGE
   AND BEARING

   IVAR CONTAINS THE LOG
   LIKELIHOOD RATIO CORRECTION
   DUE TO TRACKING

   WINDOW SIZES.
---------------------------------------
```

```
          13
-----------------------------------
   JS AND IS STORE THE ARRAY
   LOCATION OF LARGE LOG
   LIKELIHOOD RATIO

   TRACKS FOR SPECIAL PROCESSING.

   IBUFF IS CORE AREA FOR INPUT
   DATA.

   ID CONTAINS TAPE AND PING
   INFORMATION.

   INITIALIZE STORAGE
-----------------------------------


          20
      **********************
     /                      \
    /  CALL EQDISK(5,32000)  \
    \                        /
     \                      /
      **********************
                :
                :
          21    V
       *********************
       *                   *
       *  RAD=0174533      *
       *                   *
       *  NW=6             *
       *                   *
       *  KKMAX=5052-NW    *
       *                   *
       *  JJMAX=60096-NW   *
       *                   *
       *  JSTART=0         *
       *                   *
       *  B=JJMAX*05       *
       *                   *
       *  A=9*JJMAX/20000  *
       *                   *
       *********************
                :
                :
                :
                V
```

```
                          :
                          :
                   28     V
                   *************
                   *           *
                   *  KDR=200  *
                   *           *
                   *************
                          :
                          :
                   29     V
               ----------------------
                 READ INPUT DATA
               ----------------------
 / \                      :
( 160 )---------------->:
 \ /               30     V
            *****************************
           /  READ                      /
          :   900,IAA,IBB,ITHRES,ITBR,  :
         /   LIKMAX,TDB,SCLE           /
        *****************************
                          :
                          :
                   31     V
        ------------------------------------
          IAA,IBB - SCALED SLOPE AND
          INTERCEPT OF LOGLIKELIHOOD
          RATIO.

          ITHRES - SCALED LOWER DECISION
          THRESHOLD.

          ITBR - SCALED UPPER THRESHOLD
          ABOVE WHICH TRACKS ARE
          SUBJECTED

          TO SPECIAL PROCESSING.

          LIKMAX - MAXIMUM ALLOWABLE LOG
          LIKELIHOOD RATIO.

          TDB - INPUT DATA THRESHOLD IN
          DB.
        ------------------------------------
                          :
                          :
                          :
```

```
                :
                :
       37       V
 ---------------------------------
   SCLE - SCALE FACTOR FOR LOG
   LIKELIHOOD RATIO.

   OUTPUT OPTIONS  1 - SELECT, 0
   - NOT SELECT

   IPRNT - PRINT STATUS FILE

   ITOUT - OUTPUT TO MAGNETIC
   TAPE

   LUN - LOGICAL UNIT NUMBER FOR
   TAPE OUTPUT.
 ---------------------------------
                :
                :
       42       V
               .*.
            .*'    '*.     YES                              /     \
         .*'  IAA.LE.0  '*.-------------------------------->( 400 )    452
         '*.             .*'                                \     /
            '*.       .*'
               '*'
                : NO
                :
       43       V
      *************************
     /                        /
    / READ                   :
   :  900,IPRNT,ITOUT,LUN,   /
  /  ISEQ,ITIN              /
 *************************
                :
                :
       44       V
   ****************************
   *                          *
   *  THRESA=10.**(TDB/20.)   *
   *                          *
   *  SHRESA=THRESA           *
   *                          *
   ****************************
                :
                :
                :
                V
```

```
                    :
                    :
           46       V
    ---------------------------------
      INITIALIZE TRACKING WINDOWS
    ---------------------------------
                    :
                    :
           47       V
       ****************************
      /                           /
     /  READ                      :
    :   901,NVR(1),NVB(1),NVR(4) /
   /    ,NVB(4),INDS,INDL       /
  ****************************
                    :
                    :
           48       V
    ********************************
    *                              *
    *  NT=INDS                     *
    *                              *
    *  IVAR(1)=ALOG(FLOAT(NT))*SCLE *
    *                              *
    ********************************
                    :
                    :
              50    V
           *************
          /             \            54
         /  DO 150 I=2,3 \< . . . . . .
         \              /
          \            /
           *************
                    :
                    :
              51    V
        **********************
        *                    *
        *  NVR(I)=2*NVR(I-1) *
        *                    *
        *  NVB(I)=2*NVB(I-1) *
        *                    *
        *  NT=2*NT           *
        *                    *
        **********************
                    :
                    :
                    :
                    V
```

```
                            :
                            :
                            :
     /  \                   :
    ( 150 )---------------->:
     \  /              54   V
         ********************************
         *                              *
         *  IVAR(I)=ALOG(FLOAT(NT))*SCLE *
         *                              *
         ********************************
                            :
                            :
                            : . . . . . >(PAGE   5)
                            :
                       55   V
        **********************************
        *                                *
        *  IVAR(4)=ALOG(FLOAT(INDL))*SCLE *
        *                                *
        *  IVAR(5)=ALOG(FLOAT(INDL*2))*   *
        *  SCLE                          *
        *                                *
        *  NVR(5)=2*NVR(4)               *
        *                                *
        *  NVB(5)=2*NVB(4)               *
        *                                *
        *  NVRH=NVR(1)/2                 *
        *                                *
        *  NVBH=NVB(1)/2                 *
        *                                *
        *  KOLD=1                        *
        *                                *
        *  JOLD=1+JSTART                 *
        *                                *
        *  JNEW=JJMAX+JSTART+NW+1        *
        *                                *
        *  KNEW=KKMAX+NW+1               *
        *                                *
        *  KNT=KNEW                      *
        *                                *
        *  JNT=JNEW                      *
        *                                *
        *  XG=0.                         *
        *                                *
        *  YG=0.                         *
        *                                *
        **********************************
                            :
                            :
                            V
```

```
                 :
                 :
        71       V
                .*.
             .*'   '*.
          .*'         '*.
       .*'  ITOUT.LT.1.AND.  '*.YES
       I    ITIN.LT.1.OR.ISEQ  I------------------------------->( 144 )   86
       '*.  .LT.1            .*'
          '*.             .*'
             '*.       .*'
                '*'
                 : NO
                 :
        72       V
    ********************************
   /                                \
  /  CALL FINDSE(LUN,ISTAT,ISEQ,IC)   \
  \                                   /
   \                                 /
    ********************************
                 :
                 :
        73       V
                .*.
             .*'   '*.    YES
          .*' ISTAT.GT.0 '*.------------------------------------->( 300 )   450
          '*.            .*'
             '*.       .*'
                '*'
                 : NO
                 :
        74       V
             **********
             *        *
             *  NT=1  *
             *        *
             **********
                 :
( 146 )---------->:
        75       V
    ********************************
   /                                \
  /  CALL INPUTA(LUN,ISTAT,KKMAX+NW    \
  :  ,1,0,NT,NT+ID(1)-1,1,ISU(KNE+)   :
  \  )                                /
   \                                 /
    ********************************
                 :
                 :
                 V
```

```
                    :
                    :
             76     V
          *****************
          *               *
          *  NT=NT+ID(1)  *
          *               *
          *****************
                    :
                    :
             77     V
                   .*.
                .*'    '*.       YES
             .*'  ISTAT.GT.0 '*.---------------------------------->( 300 )   450
             '*.              .*'
                '*.        .*'
                   '*'
                    : NO
                    :
             78     V
                   .*.
                .*'     '*.
            .*'               '*.     YES
         .*'   ID(1).GE.ID(118)   '*.------------------------------->( 145 )   82
         '*.                      .*'
            '*.               .*'
                '*.       .*'
                   '*'
                    : NO
                    :
             79     V
      ***********************************
     /                                   \
    /  CALL DRUM(ID(1),ISU(KNEW),JNT,     \
    \  1)                                 /
     \                                   /
      ***********************************
                    :
                    :
                    :
                    :
                    :
                    :
                    V
```

```
                      :
                      :
             80       V
                     .*.
                 .*'     '*.
              .*'           '*.    YES
         .*'   JNT-JNEW.LT.ID(118  '*.------------------------------>( 146 )    75
         '*.   )                  .*'
              '*.           .*'
                 '*.     .*'
                     '*'
                      : NO
                      :
              81      V
            *********************
            *                   *
            *  JNT=JNEW+ID(118) *
            *                   *
            *********************
                        :
 ( 145 )--------------->:
               82       V
              ******************
              *                *
              *  KNT=KNT+ID(1) *
              *                *
              ******************
                        :
                        :
                83      V
                       .*.
                   .*'     '*.     YES
               .*'  ITIN.GT.0  '*.------------------------------->( 144 )    86
               '*.             .*'
                   '*.     .*'
                       '*'
                        : NO
                        :
                84      V
                *************
                *           *
                *  KNT=KNEW *
                *           *
                *  JNT=JNEW *
                *           *
                *************
                        :
                        :
                        :
                        V
```

```
                      :
                      :
                      :
 / \                  :
( 144 )------------->:
 \ /                  :
               87     V
        *****************************
       /                           /
      /  PRINT                    /
     :   906,IAA,IBB,ITHRES,ITHR,  :
     :   LIKMAX,TDB,SCLE,THRESA    :
     :                             :
     :   PRINT 907                 :
     :                             :
     :   PRINT                     :
     :   902,(NVR(I),NVB(I)        :
     :   ,IVAR(I),I=1,5)           :
     :                             :
     :   PRINT 902,NVRH,NVBH       :
     :                             :
     :   PRINT                     :
     :   909,IPRNT,ITOUT,LUN,      :
    /   ISEQ,ITIN                 /
   /                             /
   *****************************
                      :
                      :
               99     V
     ------------------------------
       INITIALIZE TAPE ROUTINE AND
       COUNTERS FOR RUN
     ------------------------------
                      :
                      :
              100     V
     *******************************
    /                               \
   /  CALL SEQCAR(ITAPE,NPING,NPMAX, \
  :   ITHSEQ,IROM,ITO,ITH,ITYPF,      :
   \  NCHAN,ISEARC,ID,IEOF)          /
    \                               /
     *******************************
                      :
                      :
                      :
                      :
                      V
```

```
                                   :
                                   :
                      102          V
                                  .*.
   /   \                       .*'   '*.
  ( 300 )<-------------.*'     IEOF     '*.------------------------------------->( 160 )   30
   \   /                 '*.         .*'                                            \   /
    450                     '*.   .*'
                               '*'
   /   \                        :  0
  ( 140 )---------------------->:
   \   /              103       V
                  ****************************
                 /                          /
                /  PRINT                   /
               :   908,ITAPE,NPING,ITHSEQ,  :
               :   NPMAX,IROM,ITH,ITO,      :
               :   ITYPE,NCHAN,             :
              /  ISEARCH,ID(1),ID(2)       /
             /                            /
            ****************************
                                :
                                :
                      108       V
              --------------------------
               INITIALIZE COUNTERS AND
               STORAGE FOR PING CYCLE.
              --------------------------
   /   \                        :
  ( 100 )---------------------->:
   \   /              109       V
                   ********************
                   *                  *
                   *  NT=KNEW         *
                   *                  *
                   *  KNEW=KOLD       *
                   *                  *
                   *  KOLD=NT         *
                   *                  *
                   *  NT=JNEW         *
                   *                  *
                   *  JNEW=JOLD       *
                   *                  *
                   *  JOLD=NT         *
                   *                  *
                   *  KMAX=KNEW+*KMAX *
                   *                  *
                   ********************
                                :
                                :
                                V
```

```
                  :
                  :
        116       V
       ********************
       *                  *
       *  JMAX=JNEW+JJMAX *
       *                  *
       *  JHP=JNT         *
       *                  *
       *  JNT=JNEW        *
       *                  *
       *  JCNT=JOLD       *
       *                  *
       ********************
                  :
                  :
        120       V
                 .*.
              .*'   '*.
          .*'           '*.      YES                            / \
       .*'    JHP.LE.JOLD    '*.------------------------------>( 110 )   126
       '*.                   .*'                                \ /
          '*.           .*'
              '*.   .*'
                 '*'
                  :  NO
                  :
        121       V
        ***************
        *             *
        * I=KKMAX+NW  *
        *             *
        ***************
                  :
                  :
        122       V
                 .*.                                        ***************
              .*'   '*.                                     *             *
          .*'           '*.      YES                        * I=JHP-JOLD  *
       .*'  JHP-JOLD.LT.I    '*.--------------------------->*             *
       '*.                   .*'                            ***************
          '*.           .*'                                        :
              '*.   .*'                                            :
                 '*'                                               :
                  :  NO                                            V
                  :<-----------------------------------------------
                  :
                  :
                  :
                  V
```

```
                          :
                 123      V
           ******************************
          /                              \
         /  CALL DRUM(I,ISU(KOLD),JCNT,0)  \
         \                                /
          \                              /
           ******************************
                          :
                 124      V
               *******************
               *                 *
               *  KHP=KOLD+I-NW  *
               *                 *
               *******************
                          :
                          V
                          ------------------------->:
                                                    :
  /   \                                             :
 ( 110 )----------------->:                         :
  \   /          126      V                         :
                ***************                     :
                *             *                     :
                *  KHP=KNT-NW *                     :
                *             *                     :
                ***************                     :
                          :                         V
  /   \                   :                         
 ( 115 )----------------->:<------------------------
  \   /          127      V
                 *************
                 *           *
                 *  LR=0     *
                 *           *
                 *  LB=0     *
                 *           *
                 *  LU=KOLD  *
                 *           *
                 *  LL=KOLD  *
                 *           *
                 *  KSS=KNE  *
                 *           *
                 *************
                          :
                          :
                          V
```

```
                      :
                      :
                 132  V
                *************
                *           *
                *  KNT=KNE  *
                *           *
                *  KR=KDR   *
                *           *
                *  JSM=0    *
                *           *
                *  JS(1)=0  *
                *           *
                *  IREC=1   *
                *           *
                *************
                      :
 ( 3 )--------------->:
                 137  V
          ************************
          *                      *
          *  IROM=(IREC-1)*IROM+1 *
          *                      *
          *  ITO=IROM+11999      *
          *                      *
          *  ICNT=-5             *
          *                      *
          ************************
                      :
                      :
                 140  V
       ---------------------------------
        READ INPUT DATA FOR PING CYCLE
       ---------------------------------
                      :
                      :
                 141  V
        ******************************
       / CALL SEGIN(ITAPE,NPING,ISTAT, \
       \ IBUFF)                         /
        ******************************
                      :
                      :
                      :
                      V
```

```
                 :
                 :
       142       V
                .*.
             .*'   '*.        YES
          .*'  IREC.GT.1 '*.--------------------------------->( 7 )     166
          '*.           .*'
             '*.     .*'
                '*'
                 : NO
                 :
       143       V
  *****************************
  *                           *
  *  X=ID(61)*RAD             *
  *  T=ID(29)*ID(31)*.00056258*
  *  XG=XG+T*SIN(X)           *
  *  YG=YG+T*COS(X)           *
  *  X=ID(92)*RAD             *
  *  T=XG+ID(93)*SIN(X)       *
  *  X=YG+ID(93)*COS(X)       *
  *  ID(95)=SQRT(X*X+T*T)     *
  *  ID(94)=ATAN(T/X)/RAD     *
  *                           *
  *****************************
                 :
                 :
       152       V
                .*.
             .*'   '*.        YES
          .*'   X.GE.0   '*.--------------------------------->( 5 )     155
          '*.           .*'
             '*.     .*'
                '*'
                 : NO
                 :
                 :
                 V
```

```
                        :
                        :
                 153    V
               ***********************
               *                     *
               *  ID(94)=ID(94)+180  *
               *                     *
               ***********************
                        :
                        :
                        V
                         ------------------------>:
                                                  :
    / \                                           :
   ( 5 )------------------>:                      :
    \ /                                           :
                 155       V                      :
                          .*.                     :
                       .*'   '*.     YES          :
                    .*'  T.GE.0 '*.-------------->:
                    '*.         .*'               :
                       '*.   .*'                  :
                          '*'                     :
                           : NO                   :
                           :                      :
                 156       V                      :
               ***********************            :
               *                     *            :
               *  ID(94)=ID(94)+360  *            :
               *                     *            :
               ***********************            :
                           :                      V
    / \                    :                      
   ( 6 )------------------>:<---------------------
    \ /                                         
                 157       V
               **************************
              /                         /
             /  PRINT                  /
            :   910,ID(1),ID(2),ID(4)  :
            :   ,ID(61),ID(31),(ID(I)  :
           /    ,I=91,95)             /
          /                          /
          ***************************
                           :
                           :
                           :
                           V
```

```
                 :
                 :
          160    V
       **********************
       *                    *
       *  ID(96)=XG         *
       *                    *
       *  ID(97)=YG         *
       *                    *
       *  NSRL=ID(95)-500   *
       *                    *
       *  NSRH=ID(95)+500   *
       *                    *
       *  NSBL=ID(94)-15    *
       *                    *
       *  NSBH=ID(94)+15    *
       *                    *
       **********************
                 :
  / \            :
 ( 7 )---------->:
  \ /            :
          166    V
                .*.
            .*'     '*.       +
         .*'   ISTAT   '*.---------------------------->( 300 )   450
         '*.           .*'
            '*.     .*'
                '*'
                 :  -,0
  / \            :
 ( 8 )---------->:
  \ /            :
          167    V
        *****************
        *               *
        *  ICNT=ICNT+6  *
        *               *
        *****************
                 :
                 :
          168    V
                .*.
            .*'     '*.
         .*'           '*.   YES
      .*'  ICNT.LE.ID(1)  '*.------------------------->( 2 )   173
      '*.                 .*'
         '*.           .*'
            '*.     .*'
                '*'
                 :  NO
                 :
                 :
                 V
```

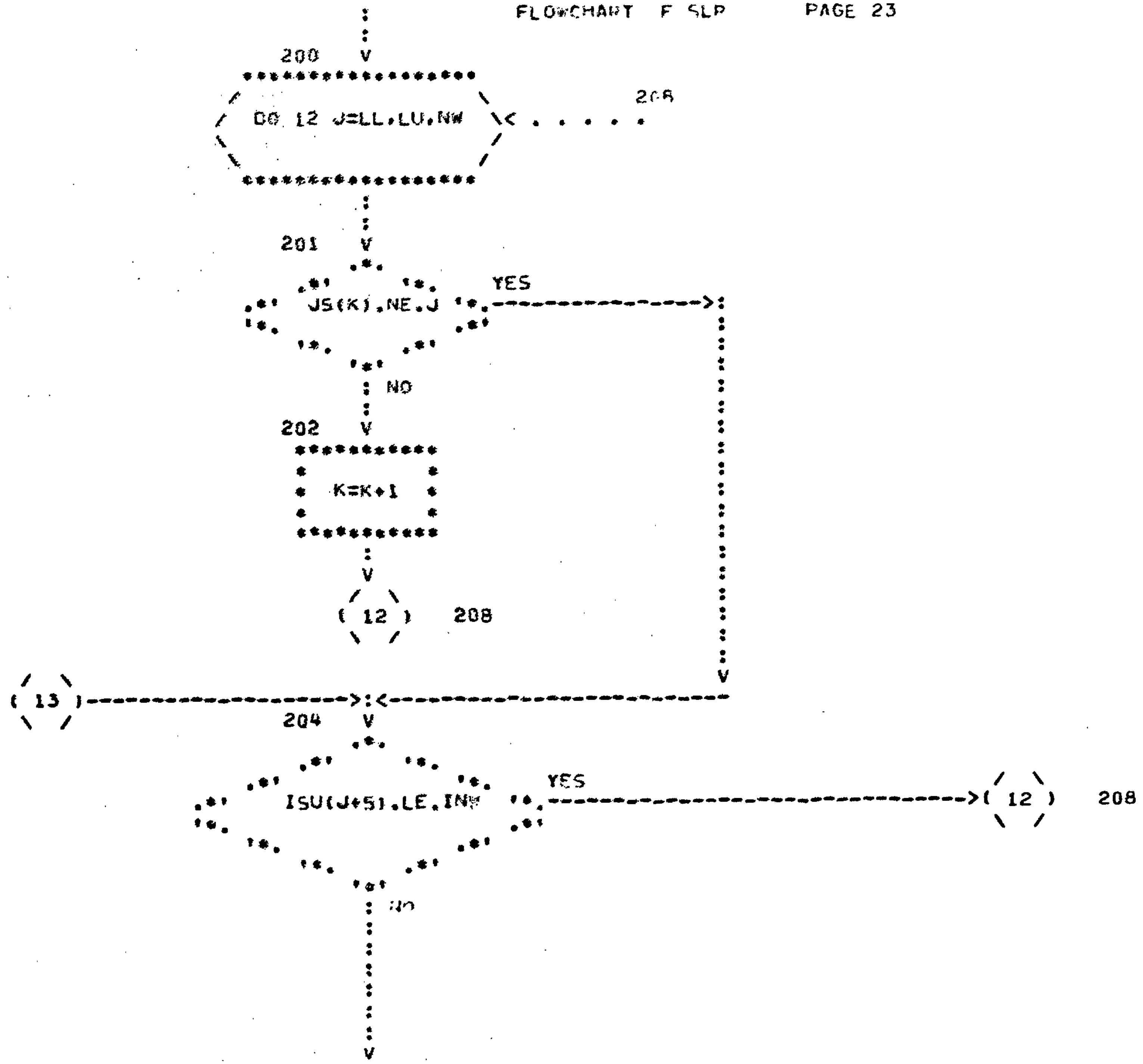
```
              :
              :
       169    V
             .*.
          .*'   '*.
       .*'         '*.     YES
    .*'  IREC.GE.ID(4)  '*.------------------------------>( 200 )   405
    '*.                 .*'
       '*.           .*'
          '*.     .*'
             '*'
              : NO
              :
       170    V
   ------------------------------------
     FIND SAMPLE LOCATION AND LOG
     LIKELIHOOD RATIO
   ------------------------------------
              :
              :
       171    V
       ****************
       *              *
       * IREC=IREC+1  *
       *              *
       ****************
              :
              V
             / \
            ( 3 )   137
             \ /

  / \
 ( 2 )------------------>:
  \ /          173       V
   ************************************
   *                                  *
   *  IR=IBUFF(ICNT)                  *
   *                                  *
   *  IB=IBUFF(ICNT+1)                *
   *                                  *
   *  T=((IBUFF(ICNT+2)+IBUFF(ICNT+5  *
   *  ))/2+IBUFF(ICNT+3)/42.-IBUFF(   *
   *  ICNT+4)/22.)/((IBUFF(ICNT+3)+   *
   *  IBUFF(ICNT+4))/64.)             *
   *                                  *
   ************************************
                         :
                         :
                         :
                         V
```

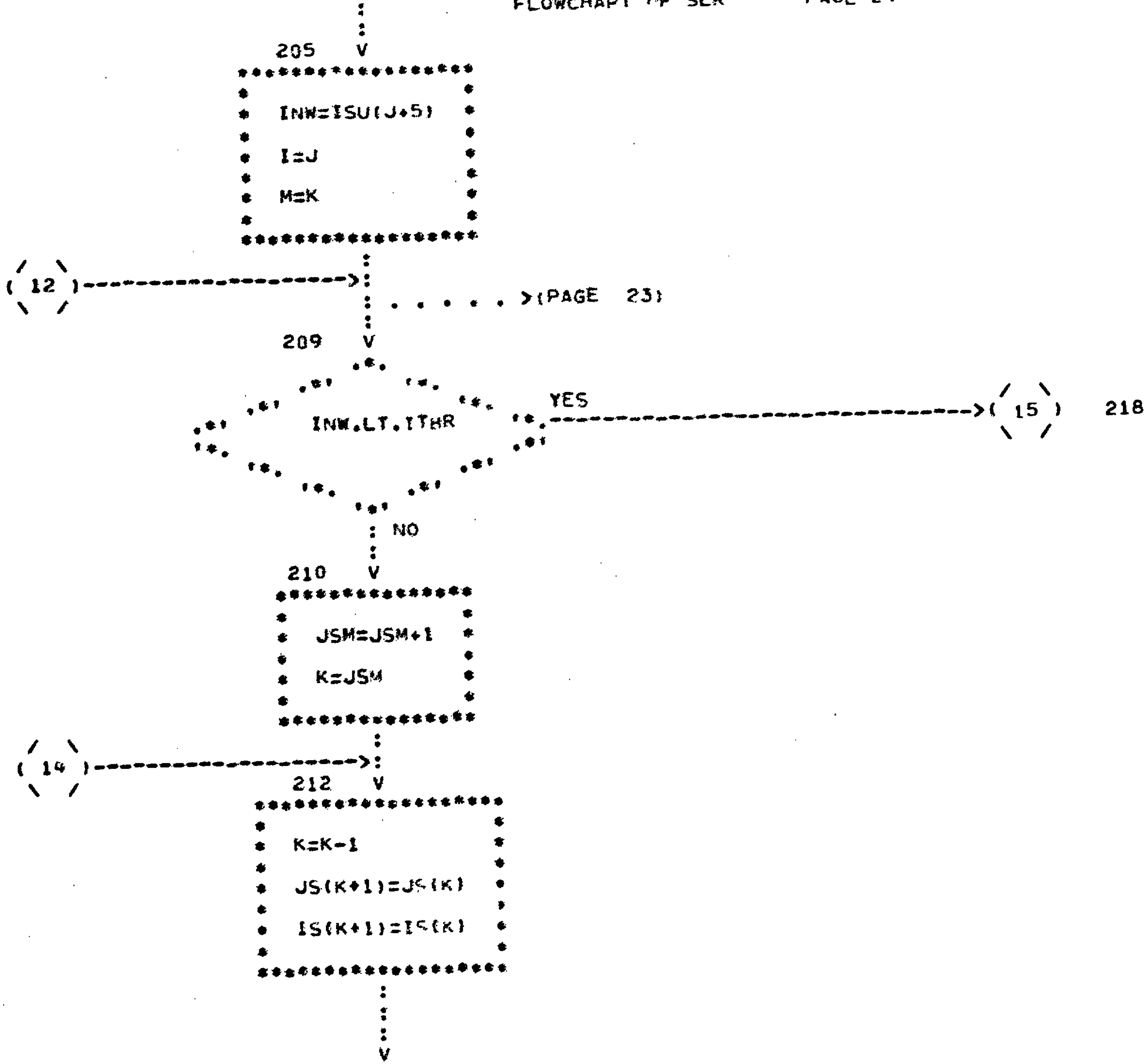
```
                   :
                   :
           177     V
                  .*.
               .*'     '*.
           .*'             '*.   YES
        .*'   T.LT.THRESA     '*.------------------------------>( 8 )   167
         '*.                 .*'
             '*.         .*'
                 '*. .*'
                   :  NO
                   :
           178     V
         ******************
         *                *
         *  LIK=IAA*T+IBB *
         *                *
         ******************
                   :
( 4 )------------->:
           179     V
                  .*.
              .*'     '*.        YES
         .*'   IR.LT.KR    '*.------------------------------>( 9 )   186
          '*.             .*'
              '*.     .*'
                  '*'
                   :  NO
                   :
           180     V
      ----------------------------
        ADAPTIVE INPUT THRESHOLD
      ----------------------------
                   :
                   :
           181     V
    ***********************************
    *                                 *
    *  DIF=FLOAT(JNT-JNEW+KNT-KNEW)/( *
    *  KR*A+B)-1.                     *
    *                                 *
    *  THRESA=THRESA*(1.+DIF*ABS(DIF) *
    *  )                              *
    *                                 *
    ***********************************
                   :
                   :
                   :
                   V
```

```
                  :
                  :
       183        V
                 .*.
              .*'     '*.
           .*'           '*.     YES
        .*'  THRESA.LT.SHRESA '*.--------------------------------->*******************
        '*.                   .*'                                  *                 *
           '*.             .*'                                     *  THRESA=SHRESA  *
              '*.       .*'                                        *                 *
                  '*'                                              *******************
                  : NO                                                     :
                  :<-------------------------------------------------------V
       184        V
      ***************
      *             *
      *  KR=KR+KOR  *
      *             *
      ***************
                  :
                  V
                 / \
                ( 4 )   179
                 \ /

 / \
( 9 )-------------------->:
 \ /                      :
       186                V
                         .*.
                      .*'   '*.
                   .*'         '*.     YES
                .*'  KHP.LE.KOLD  '*.------------------------------>( 50 )   281
                '*.               .*'                                \ /
                   '*.         .*'
                      '*.   .*'
                         '*'
                          : NO
                          :
       187                V
    ----------------------------------
     FIND LIMITS ON STORED TRACK
     INFORMATION AND UPDATE LARGE
     TRACK

     LOCATIONS.
    ----------------------------------
                          :
                          :
                          V
```

```
                          :
                          :
                189       V
              *****************
              *               *
              *  NT=LR-NVR(4) *
              *               *
              *****************
                          :
  / \                     :
 ( 10 )----------------->:
  \ /           190       V
                         .*.
                    .*'     '*.
                .*'             '*.      YES                                / \
             .*'   ISU(LL).GT.NT    '*. ---------------------------------->( 20 )     221
             '*.                    .*'                                      \ /
                '*.             .*'
                    '*.     .*'
                         '*'
                          : NO
                          :
                191       V
                         .*.
                    .*'     '*.
                .*'             '*.      YES                                / \
             .*'    LL.NE.JS(1)     '*. ---------------------------------->( 15 )     218
             '*.                    .*'                                      \ /
                '*.             .*'
                    '*.     .*'
                         '*'
                          : NO
                          :
                192       V
              ***************
              *             *
              *  JSM=JSM-1  *
              *             *
              ***************
                          :
                          :
                193       V
              ***************
             /               \                195
            /  DO 11 K=1,JSM  \<. . . . . .
            \                 /
             \               /
              ***************
                          :
                          :
                          V
```

```
                         :
                         :
               194       V
             *******************
             *                 *
             *  JS(K)=JS(K+1)  *
             *                 *
             *******************
                         :
   / \                   :
  ( 11 )---------------->:
   \ /         195       V
             *******************
             *                 *
             *  IS(K)=IS(K+1)  *
             *                 *
             *******************
                         :
                         :
                         :  . . . . . >(PAGE  21)
                         :
               196       V
              *****************
              *               *
              *  JS(JSM+1)=0  *
              *               *
              *****************
                         :
                         :
               197       V
                        .*.
                     .*'   '*.     YES                                  / \
                  .*'  JSM.LT.19  '*.--------------------------------->( 15 )   218
                  '*.             .*'                                   \ /
                     '*.       .*'
                        '*'
                         : NO
                         :
               198       V
                ***********
                *         *
                *  INW=0  *
                *         *
                *  K=1    *
                *         *
                ***********
                         :
                         :
                         :
                         :
                         V
```


200
DO 12 J=LL,LU,NW
208
201
JS(K).NE.J
YES
NO
202
K=K+1
12
208
13
204
ISU(J+5).LE.INW
YES
12
208
NO


205
INW=ISU(J+5)
I=J
M=K
12
(PAGE 23)
209
INW.LT.ITHR
YES
15
218
NO
210
JSM=JSM+1
K=JSM
14
212
K=K-1
JS(K+1)=JS(K)
IS(K+1)=IS(K)

```
                      :
                      :
              215     V
                     .*.
                 .**     **.       YES
              .**   K.GT.M   **.--------------------------->( 14 )   212
              **.              .**
                 **.       .**
                     *.*
                      : NO
                      :
              216     V
              *************
              *           *
              *  JS(M)=I  *
              *           *
              *  IS(M)=0  *
              *           *
              *************
                      :
( 15 )--------------->:
              218     V
              *************
              *           *
              * LL=LL+NV  *
              *           *
              *************
                      :
                      :
              219     V
                     .*.
                 .**     **.        YES
              .** LL.LT.KHP **.--------------------------->( 10 )   190
              **.              .**
                 **.       .**
                     *.*
                      : NO
                      :
              220     V
              ***********
              *         *
              * LL=KHP  *
              *         *
              ***********
                      :
( 20 )--------------->:
                      :
                      :
                      :
                      V
```

```
                   :
          221      V
          *****************
          *               *
          *  NT=IR+NVR(4) *
          *               *
          *****************
                   :
 ( 21 )----------->:
          222      V
                  .*.
              .*'     '*.
          .*'             '*.   YES
      .*'   ISU(LU).GT.NT    '*.---------------------------->( 30 )   245
      '*.                    .*'
          '*.             .*'
              '*.     .*'
                  '*'
                   : NO
                   :
          223      V
                  .*.
              .*'     '*.
          .*'             '*.   YES
      .*'  ISU(LU+5).LT.ITHP  '*.--------------------------->( 25 )   242
      '*.  .OR.LU.EQ.KHP      .*'
          '*.             .*'
              '*.     .*'
                  '*'
                   : NO
                   :
          224      V
                  .*.
              .*'     '*.       YES
          .*'  JSM.LT.20  '*.------------------------------->( 24 )   239
          '*.             .*'
              '*.     .*'
                  '*'
                   : NO
                   :
          225      V
          *************
          *           *
          *  I=JS(1)  *
          *           *
          *************
                   :
                   :
                   V
```

```
                 :
                 :
       226       V
    *******************
    *                 *
    *  M=1            *
    *                 *
    *  INW=ISU(I+5)   *
    *                 *
    *******************
                 :
                 :
       228       V
      ***************
     /               \                 233
    /  DO 22 K=2,JSM  \<. . . . . .
    \                 /
     \               /
      ***************
                 :
                 :
       229       V
      *************
      *           *
      *  I=JS(K)  *
      *           *
      *************
                 :
                 :
       230       V
                .*.
            .*'     '*.
        .*'             '*.       YES
    .*'   ISU(I+5).GE.INW   '*.---------------------------->( 22 )   233
    '*.                     .*'
        '*.             .*'
            '*.     .*'
                '*'
                 :  NO
                 :
       231       V
    *******************
    *                 *
    *  INW=ISU(I+5)   *
    *                 *
    *  M=K            *
    *                 *
    *******************
                 :
                 :
                 V
```

```
                    :
                    :
  / \               :
 ( 22 )------------>:
  \ /               :  . . . . . >(PAGE  27)
                    :
           234      V
                   .*.
                .*'     '*.
             .*'           '*.   YES                                 / \
       .*'   ISU(M+5).GE.ISU(LU  '*.------------------------------->( 25 )   242
       '*.   +5)                 .*'                                 \ /
             '*.           .*'
                '*.     .*'
                   '*'
                    :  NO
                    :
           235      V
         ***************
         *             *
         *  JSM=JSM-1  *
         *             *
         ***************
                    :
                    :
           236      V
         ***************
        /               \
       /  DO 23 K=M,JSM  \
       :                 :
       :  JS(K)=JS(K+1)  :
       :                 :
       \  IS(K)=IS(K+1)  /
        \               /
         ***************
  / \               :
 ( 24 )------------>:
  \ /      239      V
         ***************
         *             *
         *  JSM=JSM+1  *
         *             *
         *  JS(JSM)=LU *
         *             *
         *  IS(JSM)=   *
         *             *
         ***************
                    :
                    :
                    V
```

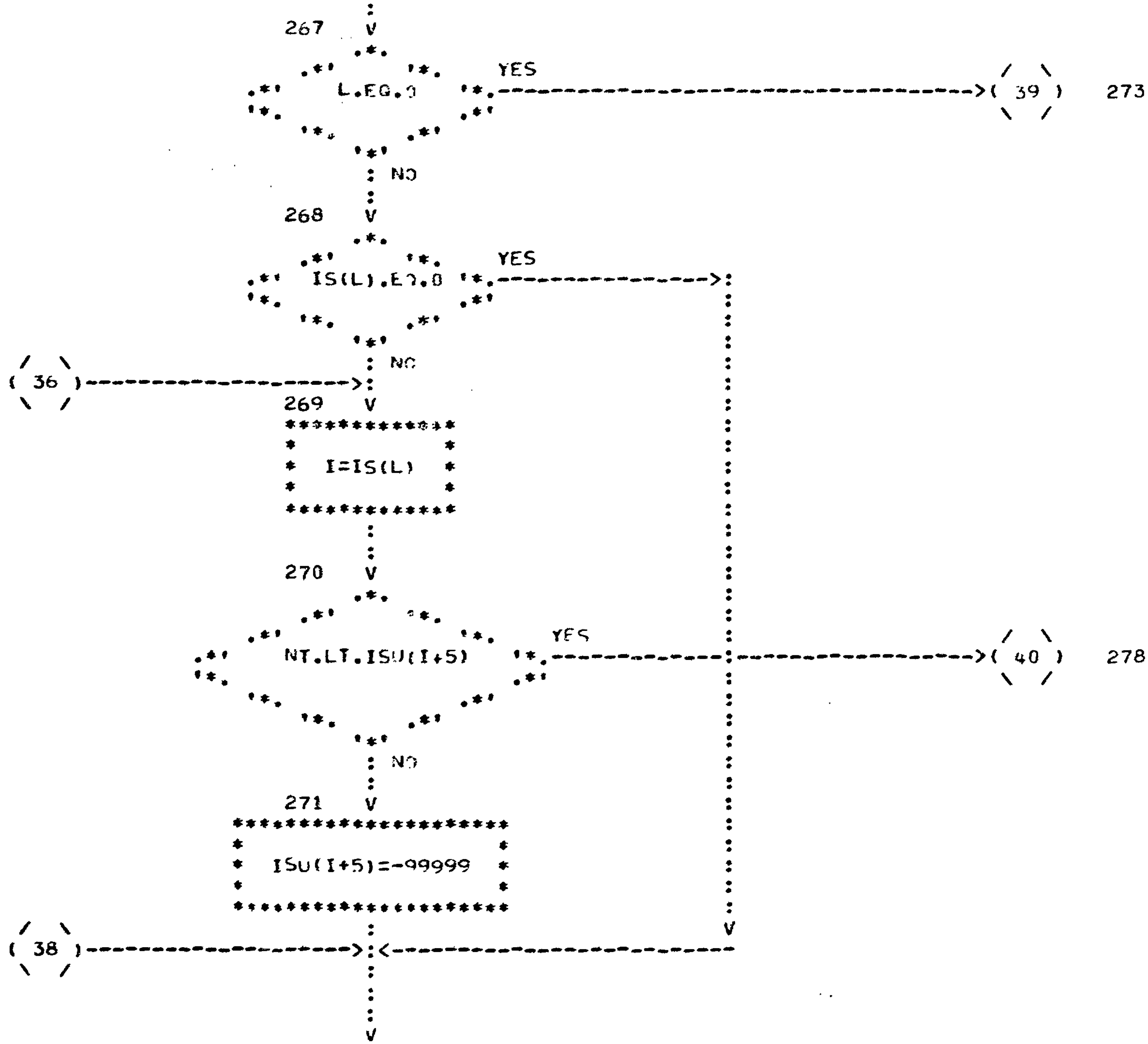
```
                     :
                     :
  / \                :
 ( 25 )------------->:
  \ /          242   V
             **************
             *            *
             *  LU=LU+NW  *
             *            *
             **************
                     :
                     :
               243   V
                    .*.
                 .*'   '*.
              .*' LU.LT.KHP '*. YES
              '*.           .*'------------------------->( 21 )    222
                 '*.     .*'
                    '*'
                     :  NO
                     :
               244   V
              ************
              *          *
              *  LU=KHP  *
              *          *
              ************
  / \                :
 ( 30 )------------->:
  \ /          245   V
                *********
                *       *
                *  K=1  *
                *       *
                *********
                     :
                     :
               246   V
            *****************
           /                 \                 278
          /  DO 40 J=LL,LU,NW \< . . . . .
          \                   /
           \                 /
            *****************
                     :
                     :
                     :
                     V
```

```
                     :
                     :
              247    V
                 *********
                 *       *
                 *  L=0  *
                 *       *
                 *********
                     :
                     :
              248    V
                    .*.
                 .*'     '*.      YES
             .*'  J.NE.JS(K)  '*.----------------->:
             '*.                .*'                 :
                 '*.       .*'                      :
                     '*'                            :
                     : NO                           :
              249    V                              :
                 *********                          :
                 *       *                          :
                 *  L=K  *                          :
                 *       *                          :
                 *********                          :
                     :                              :
                     :                              :
              250    V                              :
                    .*.                             :                   ***********
                 .*'     '*.      YES               :                   *         *
             .*'   K.LT.JSM   '*.-------------------:------------------>*  K=K+1  *
             '*.                .*'                 :                   *         *
                 '*.       .*'                      :                   ***********
                     '*'                            :                        :
                     : NO                           V                        V
  / \                :
 ( 31 )------------->:<-----------------------------------------------------
  \ /         251    V
             ******************
             *                *
             *  ISS=ISU(J+4)  *
             *                *
             ******************
                     :
                     :
              252    V
      ---------------------------------
        PROPAGATE TRACKS AS NECESSARY
      ---------------------------------
                     :
                     :
                     V
```

```
                    :
                    :
           253      V
                   .*.
                .*'     '*.
             .*'           '*.
          .*'                 '*.
       .*'  IORD(ISU(J+2),ISU(J+3),   '*.YES
      I     LR,LB).LT.0.OR.IORD(ISU(    I---------------------------->( 35 )   262
      I     J+2),ISU(J+3),IR,IB).GT.    I
       '*.  0                         .*'
          '*.                      .*'
             '*.                .*'
                '*.          .*'
                    '*'
                     :  NO
                     :
            255      V
       ********************************
       *                              *
       *  NT=ISU(J+5)+ITHRES-IVAR(ISS) *
       *                              *
       ********************************
                     :
                     :
            256      V
                    .*.
                .*'     '*.
             .*'           '*.     YES
          .*'   NT.LT.ITHRES   '*.---------------------------->( 35 )   262
          '*.                  .*'
             '*.            .*'
                '*.      .*'
                    '*'
                     :  NO
                     :
            257      V
                    .*.
                .*'     '*.
             .*'           '*.     YES
          .*'   ISS.EQ.3.OR.ISS '*.---------------------------->( 35 )   262
          '*.   .EQ.5           .*'
             '*.             .*'
                '*.       .*'
                    '*'
                     :  NO
                     :
                     :
                     :
                     V
```

```
                 :
                 :
        258      V
                .*.
            .*'     '*.
        .*'             '*.  YES
    .*'    KNT.GT.KMAX       '*.---------------------------------->( 59 )   287
    '*.                     .*'
        '*.             .*'
            '*.     .*'
                '*'
                 : NO
                 :
        259      V
  ********************************
 /                                \
/  CALL SU(ISU(KNT),ISU(J+2),ISU(  \
:  J+3),ISU(J),ISU(J+1),ISU(J+4)+  :
\  1,NT)                           /
 \                                /
  ********************************
                 :
                 :
        260      V
        ****************
        *              *
        *  KNT=KNT+NW  *
        *              *
        ****************
                 :
                 :
        261      V
    ----------------------------
      CHECK FOR TRACK LINKAGE
    ----------------------------
                 :
( 35 )---------->:
                 :
                 :
                 :
                 :
                 :
                 V
```

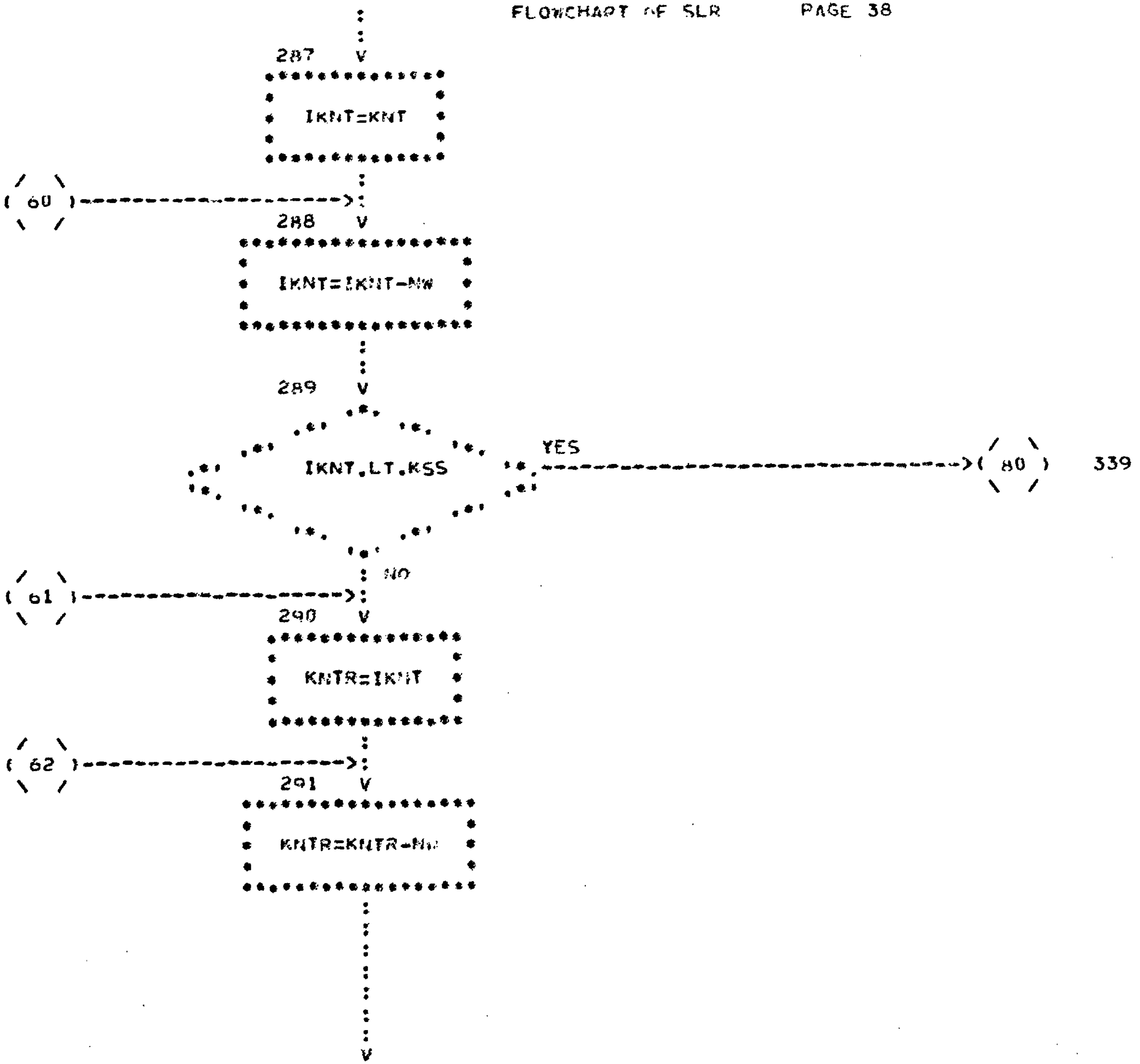
```
                :
                :
     262        V
               .*.
           .*'     '*.
       .*'             '*.
   .*'   ICOMP(IR,IE,ISU(J+2)-NVR   '*.YES
 I       (ISS),ISU(J+3)-NVB(ISS),     I-------------------------->( 40 )   278
 I       ISU(J+2)+NVR(ISS),ISU(J+     I
 '*.     3)+NVB(ISS)).NE.0          .*'
     '*.                        .*'
         '*.                .*'
             '*.        .*'
                 '*'
                  : NO
                  :
     264          V
 ------------------------------------
   CALCULATE NEW LOG LIKELIHOOD
   RATIO
 ------------------------------------
                  :
                  :
     265          V
  *******************************
  *                             *
  *  NT=ISU(J+5)+LIK-IVAR(ISS)  *
  *                             *
  *******************************
                  :
                  :
     266          V
                 .*.
             .*'     '*.
         .*'             '*.     YES
     .*'    NT.LT.ITHRES     '*.-------------------------->( 40 )   278
     '*.                     .*'
         '*.             .*'
             '*.     .*'
                 '*'
                  : NO
                  :
                  :
                  :
                  V
```


267
L.EQ.0
YES
39
273
NO
268
IS(L).EQ.0
YES
NO
36
269
I=IS(L)
270
NT.LT.ISU(I+5)
YES
40
278
NO
271
ISU(I+5)=-99999
38

```
              :
              :
     272      V
             .*.
          .*'   '*.
       .*'         '*.     YES
    .*'  KNT.LE.KMAX  '*.-------------------------------->* IS(L)=KNT *
     '*.             .*'                                  *************
        '*.       .*'                                           :
           '*. .*'                                              :
              : NO                                              V
 ( 39 )---------------->:<---------------------------------------
     273      V
             .*.
          .*'   '*.
       .*'         '*.     YES
    .*'  KNT.GT.KMAX  '*.-------------------------------->( 59 )   287
     '*.             .*'
        '*.       .*'
           '*. .*'
              : NO
              :
     274      V
             .*.
          .*'   '*.
       .*'         '*.     YES
    .*' ISU(J+5).GT.LIKMAX '*.------------------------>* NT=LIKMAX-IVAR(ISS)+LIK *
     '*.             .*'                               ****************************
        '*.       .*'                                              :
           '*. .*'                                                 :
              : NO                                                 V
              :<----------------------------------------------------
     275      V
 ------------------------------
   CREATE NEW MULTIPING TRACK
   PACKET
 ------------------------------
              :
              :
              :
              V
```

```
                  :
                  :
           276    V
        ******************************
       /                              \
      /  CALL SU(ISU(KNT),IR,IB,ISU(J),\
      \  ISU(J+1),1,NT)                /
       \                              /
        ******************************
                   :
                   :
            277    V
        ***************
        *             *
        * KNT=KNT+NW  *
        *             *
        ***************
                   :
 / \               :
( 40 )------------>:
 \ /               :  . . . . . >(PAGE  29)
                   :
             279   V
             ***********
             *         *
             *  LR=IR  *
             *         *
             *  LB=IB  *
             *         *
             ***********
                   :
 / \               :
( 50 )------------>:
 \ /         281   V
                  .*.
              .*'     '*.
          .*'             '*.     YES                            / \
       .*'    LIK.LT.ITHRES  '*.------------------------------>( 59 )   287
       '*.                   .*'                                 \ /
          '*.             .*'
              '*.     .*'
                  '*'
                   :  NO
                   :
                   :
                   :
                   :
                   V
```

```
              :
              :
       282    V
             .*.
          .*'     '*.
       .*'           '*.      YES
    .*'    KNT.GT.KMAX    '*.------------>:
    '*.                   .*'             :
       '*.             .*'                :
          '*.       .*'                   :
             '*'                          :
              : NO                        :
              :                           :
       283    V                           :
 ----------------------------             :
    CREATE SINGLE PING TRACK              :
    PACKET                                :
 ----------------------------             :
              :                           :
              :                           :
       284    V                           :
 ******************************           :
 /                            \           :
/  CALL SU(ISU(KNT),IR,IB,IR,IB,4 \       :
\  ,LIK)                      /           :
 \                            /           :
 ******************************           :
              :                           :
              :                           :
       285    V                           :
       ***************                    :
       *             *                    :
       *  KNT=KNT+NW *                    :
       *             *                    :
       ***************                    :
              :                           :
              :                           :
       286    V                           :
 ------------------------------           :
   TRACK REDUNDACY REMOVAL AND            :
   ORDERING SECTION.                      :
 ------------------------------           :
 / \          :                           V
( 59 )------->:<---------------------------
 \ /          :
              :
              :
              :
              :
              :
              V
```


287
IKNT=KNT
60
288
IKNT=IKNT-NW
289
IKNT.LT.KSS
YES
80
339
NO
61
290
KNTR=IKNT
62
291
KNTR=KNTR-NW

```
                 :
                 :
      292        V
                .*.
           .*'       '*.
       .*'               '*.   YES
   .*'     KNTR.LT.KNEW      '*.------------------------------>( 60 )   288
   '*.                       .*'
       '*.               .*'
           '*.       .*'
                '*'
                 :  NO
                 :
      293        V
 ************************************
 *                                  *
 *  NT=ICOMP(ISU(KNTR),ISU(KNTR+1)  *
 *  ,ISU(IKNT)-NVRH,ISU(IKNT+1)-    *
 *  NVBH,ISU(IKNT)+NVRH,ISU(IKNT+1  *
 *  )+NVBH)                         *
 *                                  *
 ************************************
                 :
                 :
      295        V
                .*.
           .*'       '*.
       .*'               '*.   YES
   .*'  NT.LT.-1.AND.KNTR    '*.------------------------------>( 60 )   288
   '*.  .LE.KSS              .*'
       '*.               .*'
           '*.       .*'
                '*'
                 :  NO
                 :
      296        V
                .*.
           .*'       '*.   YES
       .*'   NT.GT.1     '*.---------------------------------->( 70 )   323
       '*.               .*'
           '*.       .*'
                '*'
                 :  NO
                 :
                 :
                 V
```

```
                         :
                         :
              297        V
                        .*.
                     .*'     '*.
  / \           +  .*'    NT    '*. -
 ( 73 )<-----------.*'             '*.------------------------------>( 62 )   291
  \ /                '*.         .*'                                    \ /
  302                   '*.   .*'
                           '*'
                            :  0
  / \                       :
 ( 63 )-------------------->:
  \ /                       :
               298          V
                           .*.
                       .*'     '*.
                   .*'             '*.
               .*'  ISU(IKNT+5).GT.    '*.YES
              I     ITBR.OR.ISU(KNTR+5    I------------------------->( 64 )   303
               '*.  ).GT.ITBR          .*'                              \ /
                   '*.             .*'
                       '*.     .*'
                           '*'
                            :  NO
                            :
               299          V
        *************************************
        *                                   *
        *  NT=ICOMP(ISU(KNTR+2),ISU(KNTR+   *
        *  3),ISU(IKNT+2)-NVRH,ISU(IKNT+3   *
        *  )-NVBH,ISU(IKNT+2)+NVRH,ISU(     *
        *  IKNT+3)+NVBH)                    *
        *                                   *
        *************************************
                            :
                            :
               301          V
                           .*.
                       .*'     '*.   YES
                   .*'   NT.EQ.0   '*.------------------------------>( 64 )   303
                   '*.             .*'                                  \ /
                       '*.     .*'
                           '*'
  / \                       :  NO
 ( 73 )-------------------->:
  \ /                       :
                            :
                            :
                            :
                            :
                            :
```

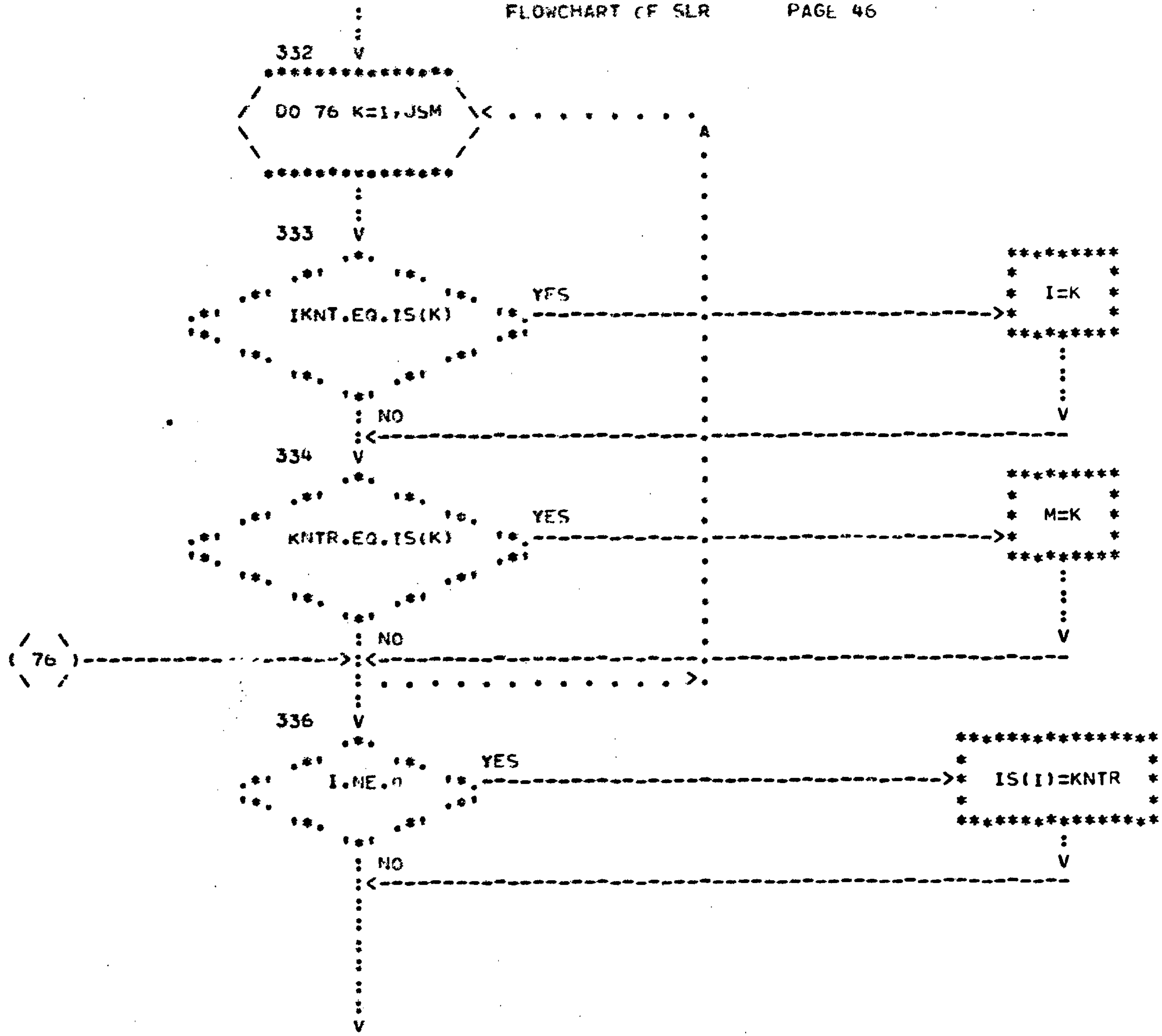
```
                    :
                    :<--------------------------------
           302      V                                A
                   .*.                               :
               .*'     '*.                           :
           .*'             '*.                       :
  / \    +  .*'  IORD(ISU(KNTR),ISU  '*.-,0          :                        / \
( 70 )<-------I  (KNTR+1),ISU(IKNT)    I-------------:---------------------->( 62 )   291
  \ /       '*.  ,ISU(IKNT+1))       .*'             :                        \ /
  323          '*.                 .*'               :
                   '*.         .*'                   :
                       '*.  .*'                      :
                                                     :
  / \                                                :
( 64 )------------------->:                          :
  \ /              303    V                          :
                         .*.                         :
                     .*'     '*.                     :
                 .*'             '*.                 :
             .*'   ISU(KNTR+4).EQ.4  '*.YES          :
             I     .OR.ISU(IKNT+4)     I----------->:
             '*.   .EQ.4             .*'
                 '*.             .*'
                     '*.     .*'
                        '*'
                         : NO
                         :
                   304   V
               ****************
               *              *
               *  KNT=KNT-NW  *
               *              *
               ****************
                         :
                         :
                   305   V
                        .*.
                    .*'     '*.
                .*'             '*.      YES                                  / \
             .*'  ISU(KNTR+5).GT.ISU '*.-------------------------------->( 69 )   313
             '*.  (IKNT+5)           .*'                                      \ /
                '*.             .*'
                    '*.     .*'
                        '*'
                         : NO
                         :
                         :
                         V
```

```
            :
            :
  306       V
  *************
  *           *
  *  JL=KNTR  *
  *           *
  *************
            :
            :
  307       V
           .*.
       .*'     '*.      YES
    .*'  JL.LT.KSS '*.----------------------------------->*   KSS=KSS-NW   *
    '*.               .*'                                  ****************
       '*.         .*'                                            :
           '*.*'                                                  V
            : NO
 ( 65 )---------------->:<-------------------------------------------
  308       V
  *************
  *           *
  *  NT=KNT-1 *
  *           *
  *************
            :
            :
  309       V
  *****************
 /  DO66 I=JL,NT   \
 :  INW=I+NW       :
 \  ISU(I)=ISU(INW) /
  *****************
            :
            V
          ( 67 )    315

 ( 69 )---------------->:
                        :
                        :
                        V
```

```
                 :
                 :
          313    V
          *************
          *           *
          *  JL=IKNT  *
          *           *
          *************
                 :
                 :
          314    V
                .*.
            .*'     '*.
        .*'             '*.     YES                                 / \
     .*'     KNT.NE.IKNT     '*.--------------------------------->( 65 )    308
     '*.                     .*'                                    \ /
        '*.             .*'
            '*.     .*'
                '*'
                 : NO
 / \             :
( 67 )---------->:
 \ /      315    V
                .*.
            .*'     '*.     YES                                     / \
        .*'   JSM.EQ.0   '*.------------------------------------->( 60 )    288
        '*.              .*'                                        \ /
            '*.     .*'
                '*'
                 : NO
                 :
          316    V
          **************
         /              \                 321
        /  DO 68 K=1,JSM  \<. . . . . .
        \                 /
         \               /
          **************
                 :
                 :
          317    V
                .*.
 / \        +   .*'     '*.    -                                    / \
( 72 )<-------.*'  IS(K)-JL  '*.--------------------------------->( 68 )    321
 \ /          '*.            .*'                                    \ /
  320             '*.     .*'
                      '*'
                       : 0
                       :
                       :
                       V
```

```
                           :
                           :
  / \                      :
 ( 71 )------------------->:
  \ /               318    V
                    *************
                    *           *
                    *  IS(K)=0  *
                    *           *
                    *************
                           :
                           :
                           V
                           ------------------------>:
                                                    :
  / \                                               :
 ( 72 )------------------->:                        :
  \ /               320    V                        :
                 *******************                :
                 *                 *                :
                 *  IS(K)=IS(K)-NW *                :
                 *                 *                :
                 *******************                :
  / \                      :                        V
 ( 68 )------------------->:<------------------------
  \ /                      : . . . . . >(PAGE  43)
                           V
                          / \
                         ( 60 )   288
                          \ /

  / \
 ( 70 )------------------->:
  \ /               323    V
                   **************
                  /              \         328
                 /  DO 75 I=1,N*  \< . . . . . .
                 \                /
                  \              /
                   **************
                           :
                           :
                           :
                           :
```

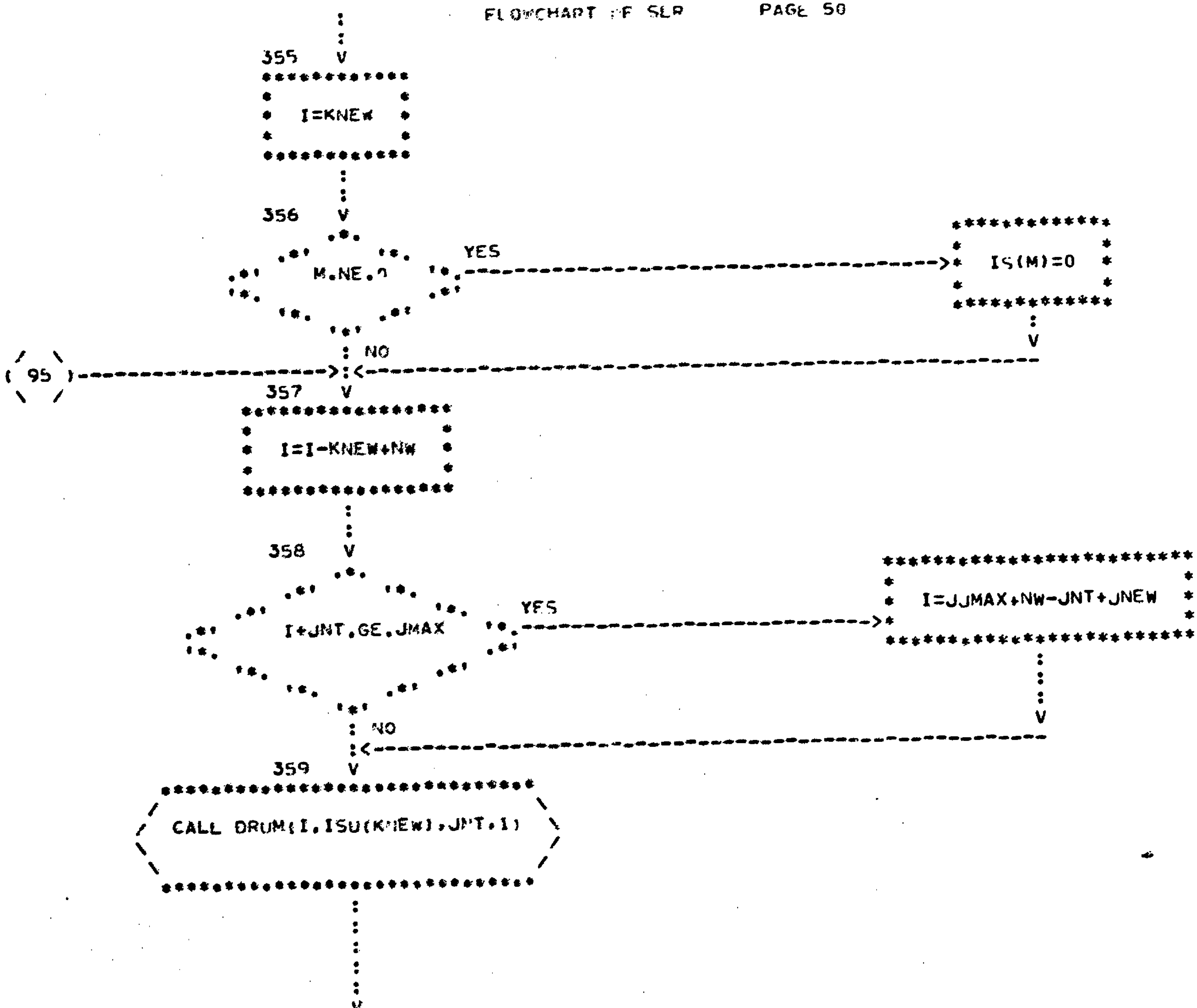
```
                        :
                        :
              324       V
               *********************
               *                   *
               *  INW=KNTR+I-1     *
               *                   *
               *  NT=ISU(INW)      *
               *                   *
               *  M=IKNT+I-1       *
               *                   *
               *  ISU(INW)=ISU(M)  *
               *                   *
               *********************
                          :
  ( 75 )----------------->:
               328        V
                 **************
                 *            *
                 *  ISU(M)=NT *
                 *            *
                 **************
                          :
                          :  . . . . . >(PAGE  44)
                          :
               329        V
                         .*.
                     .*'     '*.   YES
                 .*'  JSM.EQ.0   '*.-------------------------------->( 61 )   290
                 '*.             .*'
                     '*.     .*'
                         '*'
                          : NO
                          :
               330        V
                    *********
                    *       *
                    *  I=0  *
                    *       *
                    *  M=0  *
                    *       *
                    *********
                        :
                        :
                        V
```


332
DO 76 K=1,JSM
333
IKNT.EQ.IS(K)
YES
I=K
NO
334
KNTR.EQ.IS(K)
YES
M=K
NO
76
336
I.NE.0
YES
IS(I)=KNTR
NO

```
                  :
                  :
       337        V
                 .*.
              .*'     '*.        YES
           .*'   M.NE.0   '*.-------------------------------->* IS(M)=IKNT  *
            '*.           .*'                                 ****************
              '*.       .*'                                          :
                 '*'                                                  V
                  : NO
                  V<-------------------------------------------------
                 / \
                ( 61 )   290
                 \ /

  / \
 ( 80 )----------------->:
  \ /             339    V
                 *************
                 *           *
                 *  KSS=KNT  *
                 *           *
                 *************
                       :
                       :
                 340   V
          -------------------------------
            DISK INPUT/OUTPUT OF TRACK
            INFORMATION
          -------------------------------
                       :
                       :
                 341   V
                      .*.
                  .*'     '*.
              .*'             '*.     YES                           / \
           .*'  KNT+300.LT.KMAX  '*.------------------------------>( 191 )   378
            '*.                 .*'                                 \ /
               '*.           .*'
                  '*.     .*'
                      '*'
                       : NO
  / \                  :
 ( 99 )--------------->:
  \ /                  :
                       :
                       :
                       V
```

```
                          :
                    342   V
                    ************ ***
                    *                *
                    *  NT=KNT-NW     *
                    *                *
                    *  I=NT          *
                    *                *
                    ****************
                          :
 ( 98 )------------------>:<--------------------------
                    344   V                          A
                    ************                     :
                    *          *                     :
                    *  I=I-NW  *                     :
                    *          *                     :
                    ************                     :
                          :                          :
                    345   V                          :
                         .*.                         :
                     .*'     '*.                     :
                  .*'           '*.  YES             :
               .*' ISU(I).LT.ISU(NT)- '*.------------>:  :
               '*. NVRH              .*'          :  :
                  '*.             .*'             :  :
                     '*.       .*'                :  :
                        '*'                       :  :
                          : NO                    :  :
                    346   V                       :  :
                         .*.                      :  :
                     .*'     '*.    YES           :  :
                  .*'  I.GT.KNEW '*.----------------->:
                  '*.            .*'              :
                     '*.      .*'                 :
                        '*'                       V
                          : NO                    :
 ( 91 )------------------>:<-----------------------
                    347   V
                         .*.
                     .*'     '*.    YES
                  .*'  JSM.EQ.0  '*.-------------------------->( 95 )   357
                  '*.            .*'
                     '*.      .*'
                        '*'
                          : NO
                          :
                          V
```

```
                :
                :
        348     V
             *********
             *       *
             *  M=0  *
             *       *
             *********
                :
                :
        349     V
           ***************
          /               \
         /  DO 92 K=1,JSM  \<. . . . . . . . . .
         \                 /                    A
          \               /                     .
           ***************                      .
                :                               .
        350     V                               .
               .*.                              .
            .*'     '*.    YES                  .
         .*'  I.LT.IS(K) '*.--------------->:   .
         '*.               .*'              :   .
            '*.         .*'                 :   .
               '*'                          :   .
                :  NO                       :   .
        351     V                           :   .
           ***************                  :   .
           *             *                  :   .
           *  I=IS(K)-NW *                  :   .
           *             *                  :   .
           *  M=K        *                  :   .
           *             *                  :   .
           ***************                  V   .
  / \           :                               .
 ( 92 )-------->:<------------------------------.
  \ /           : . . . . . . . . . . . . . . .>.
                :
        354     V
               .*.
            .*'     '*.    YES                           / \
         .*'  I.GT.KTEM  '*.------------------------->( 95 )   357
         '*.               .*'                           \ /
            '*.         .*'
               '*'
                :  NO
                :
                V
```


355
I=KNEW
356
M.NE.0
YES
IS(M)=0
NO
95
357
I=I-KNEW+NW
358
I+JNT.GE.JMAX
YES
I=JJMAX+NW-JNT+JNEW
NO
359
CALL DRUM(I,ISU(KNEW),JNT,1)

```
          :
          :
   360    V
 ****************
 *              *
 *  NT=KNEW+I-1 *
 *              *
 ****************
          :
          :
   361    V
         .*.
      .*'   '*.     YES
   .*' IPRNT.EQ.1 '*.---------------------->/  PRINT                       /
   '*.             .*'                    /  903,(ISU(K),K=KNEW,NT)     /
      '*.       .*'                      *****************************
         '*.*'                                        V
          : NO                                        :
          :<------------------------------------------
   363    V
 *********************
/                     \
/  DO 122 K=KNEW,NT,NW \< . . . .  367
\                     /
 *********************
          :
          :
   364    V
         .*.
      .*'   '*.
   .*'         '*.
.*'  ICOMP(ISU(K),ISU(K '*.YES
I    +1),NSRL,NSBL,NSRH   I------------------------>( 122 )   367
'*.  ,NSBH).NE.0        .*'
   '*.               .*'
      '*.         .*'
         '*.   .*'
          : NO
          :
   365    V
 ****************
 *              *
 *  INW=K+NW-1  *
 *              *
 ****************
          :
          :
          :
```

```
                         :
                         :
                  366    V
           ************************
          /  PRINT                /
         /  903,(ISU(L),L=K,INW)  /
        /                        /
       ************************
   / \                   :
  ( 122 )---------------->:
   \ /                   :  . . . . . >(PAGE  51)
                         :
                  368    V
                        .*.
                     .*'   '*.
                  .*'         '*.       YES
               .*'  JNT.EQ.JMAX  '*.---------------------------->( 300 )    450
               '*.               .*'
                  '*.         .*'
                     '*.   .*'
                        '*'
                         : NO
                         :
                  369    V
                *************
                *           *
                * KNT=KNT-I *
                *           *
                * KSS=KSS-I *
                *           *
                * NT=KNT-1  *
                *           *
                *************
                         :
                         :
                  372    V
               ***************
              /               \
             / DO 93 J=KNFW,NT \
             :                 :
             : INW=J+I         :
             :                 :
             \ ISU(J)=ISU(INW) /
              \               /
               ***************
                         :
                         :
                         :
                         V
```

```
                            :
                    375     V
                  ****************
                 /                \
                /  DO 190 K=1,JSM  \< . . . . . . . . . .
                \                  /                     A
                 \                /                      .
                  ****************                       .
                            :                            .
                            :                            .
                    376     V                            .
                           .*.                           .              ******************
                        .*'   '*.    YES                 .              *                *
                     .*' IS(K).NE.0 '*.------------------.------------->*  IS(K)=IS(K)-I  *
                     '*.             .*'                 .              *                *
                        '*.       .*'                    .              ******************
                           '*'                           .                      :
                            : NO                         .                      V
   / \                      :                            .
 ( 190 )------------------->:<---------------------------.-----------------------
   \ /                      :  . . . . . . . . . . . . . >.
                            :
                            :
   / \                      :
 ( 191 )------------------->:
   \ /              378     V
                           .*.
                        .*'   '*.    YES                                  / \
                     .*'  LU.LT.KHP '*.------------------------------->( 8 )    167
                     '*.             .*'                                  \ /
                        '*.       .*'
                           '*'
                            : NO
                            :
                    379     V
                           .*.
                        .*'   '*.
                     .*'         '*.     YES                              / \
                  .*'   JCNT.GE.JHP  '*.--------------------------------->( 8 )    167
                  '*.                .*'                                  \ /
                     '*.          .*'
                        '*.    .*'
                           '*'
                            : NO
                            :
                            :
                            :
```

```
               :
               :
      380      V
              .*.
           .*'   '*.    YES
       .*'  LL.EQ.KOLD '*.------------------------------------>*  LL=KOLD+NW  *
       '*.                .*'                                   ****************
           '*.        .*'                                              :
              '*'                                                      V
               : NO
               :<------------------------------------------------------
      381      V
     ******************
     *                *
     *  I=LL-KOLD     *
     *                *
     *  NT=KHP-I+NW-1 *
     *                *
     ******************
               :
               :
      383      V
     *****************
    /                 \
   /  DO 192 K=KOLD,NT \
   :                   :
   :  INW=K+I          :
   :                   :
   \  ISU(K)=ISU(INW)  /
    \                 /
     *****************
               :
               :
      386      V
       *************
       *           *
       *  LL=LL-I  *
       *           *
       *  LU=LU-I  *
       *           *
       *************
               :
               :
               :
               V
```

```
                  :
                  :
         388      V
                 .*.
              .*'     '*.        YES
          .*'   JSM.EQ.0   '*.------------------------------------->(  193  )    399
          '*.                .*'
              '*.         .*'
                  '*'
                  : NO
                  :
         389      V
                 .*.
              .*'     '*.
          .*'                '*.      YES
       .*'   JS(1)-I.GE.LL      '*.------------------------------->(  194  )    397
       '*.                      .*'
          '*.                .*'
              '*.         .*'
                  '*'
                  : NO
                  :
         390      V
        ***************
        *             *
        *  JSM=JSM-1  *
        *             *
        ***************
                  :
                  :
         391      V
        *****************
       /                 \
      /  DO 196 K=1,JSM   \
      :  IS(K)=IS(K+1)    :
      \  JS(K)=JS(K+1)-I  /
       \                 /
        *****************
                  :
                  :
         394      V
        *****************
        *               *
        *  IS(JSM+1)=0  *
        *               *
        *****************
                  :
```

```
                 :
         395     V
        ****************
        *              *
        *  JS(JSM+1)=0 *
        *              *
        ****************
                 :
                 V
                 ------------------------>:
                                          :
( 194 )---------------->:                 :
         397     V                        :
        ****************                  :
       /                \                 :
      /  DO 195 K=1,JSM  \                :
      :                  :                :
      \  JS(K)=JS(K)-1   /                :
       \                /                 :
        ****************                  V
( 193 )---------------->:<-----------------
         399     V
         *************
         *           *
         *  NT=NT+1  *
         *           *
         *************
                 :
         400     V
                .*.
            .*'     '*.
         .*'           '*.
      .*'  .GT.JHP-JCNT+NW '*.  YES              ****************
      '*.                 .*' ----------------->*  I=JHP-JCNT   *
         '*.           .*'                      ****************
            '*.     .*'                                :
                '*'                                    V
                 : NO                                  
                 :<-------------------------------------
                 :
                 V
```

```
                 :
                 :
         401     V
        ****************
        *              *
        *  KHP=NY+I-NW *
        *              *
        ****************
                 :
                 :
         402     V
    ***************************
   /                           \
  /  CALL DRUM(I,ISU(NT),JCNT,0) \
  \                             /
   \                           /
    ***************************
                 :
                 V
                / \
               ( 8 )    167
                \ /

         404
    ------------------------------
     END OF PING CYCLE CLEAN-UP
    ------------------------------

  / \
 ( 200 )----------------->:
  \ /    405     V
                .*.
             .*'     '*.
          .*'           '*.   YES
       .*'   JNT.EQ.JNEW    '*.------------------------------>( 220 )   409
       '*.                  .*'
          '*.           .*'
             '*.     .*'
                '*'
                 : NO
                 :
         406     V
        ***************
        *             *
        * I=KNT-KNEW  *
        *             *
        ***************
                 :
                 :
                 V
```

```
                 :
                 :
        407      V
                .*.
             .*'   '*.
          .*'         '*.       YES
       .*'  I+JNT.GT.JMAX  '*.------------------------------>* I=JMAX+NW-JNT *
       '*.                 .*'                                         :
          '*.         .*'                                              :
             '*.   .*'                                                 :
                '*'                                                    V
                 : NO                                                  
                 :<-----------------------------------------------------
        408      V
      *****************************
     /  CALL DRUM(I,ISU(KNEW),JNT,1)  \
     \                              /
      *****************************
                 :
 ( 220 )-------->:
        409      V
        **********************
        *                    *
        *  MT=(KNT-KNEW)/NW  *
        *                    *
        *  NT=(JNT-JNEW)/NW  *
        *                    *
        **********************
                 :
                 :
        411      V
        **************************
       /  PRINT 911,NPING,MT,NT  /
      /                         /
      *************************
                 :
                 :
        413      V
           *************
           *           *
           *  I=KNT-1  *
           *           *
           *************
                 :
                 :
                 :
                 V
```

```
                  :
                  :
          414     V
                 .*.
              .*'   '*.      YES
          .*'  IPRNT.EQ.1  '*.------------------------------>/  PRINT                 /
          '*.              .*'                              /   903,(ISU(K),K=KNEW,I) /
              '*.       .*'                                *************************
                  '*.*'                                               V
                   : NO                                               :
                   :<-------------------------------------------------
          415      V
           ********************          419
          / DO 222 K=KNEW, I,NW  \< . . . .
          \                      /
           ********************
                   :
                   :
          416      V
                  .*.
              .*'     '*.
          .*'  ICOMP(ISU(K),ISU(K  '*.YES
          I    +1),NSRL,NSRL,NSRH    I-------------------------->( 222 )   419
          '*.  ,NSRH).NE.0         .*'
              '*.              .*'
                  '*.       .*'
                     '*.*'
                      : NO
                      :
          417         V
           ***************
           *             *
           *  INW=K+NW-1 *
           *             *
           ***************
                  :
                  :
          418     V
           *************************
          / PRINT                  /
         /  903,(ISU(L),L=K,INW)  /
           *************************
                  :
                  :
                  V
```

```
                       :
   / \                 :
  ( 222 )------------->:
   \ /                 :
                       :  . . . . . >(PAGE  59)
                 420   V
                      .*.
                   .*'   '*.      -,0                      / \
                .*'  ITOUT  '*.----------------------->( 240 )   448
                '*.         .*'                            \ /
                   '*.   .*'
                      '*'
   / \                 :
  ( 225 )------------->:
   \ /           421   V
                      .*.
                   .*'   '*.
                .*'          '*.   YES                     / \
             .*'  JNT.EQ.JNEW   '*.------------------->( 230 )   433
             '*.                .*'                        \ /
                '*.          .*'
                   '*.   .*'
                      '*'
                       : NO
                       :
                 422   V
            ************************
            *                      *
            *   ID(1)=KKMAX+NW     *
            *                      *
            *   ID(118)=JNT-JNEW   *
            *                      *
            ************************
                       :
                       :
                 424   V
                      .*.
                   .*'   '*.                       *****************
                .*'          '*.   YES             *               *
             .*'  ID(1).GT.ID(118) '*.------------>*  ID(1)=ID(118) *
             '*.                .*'                *               *
                '*.          .*'                   *****************
                   '*.   .*'                               :
                      '*'                                  :
                       : NO                                V
                       :<----------------------------------
                       :
                       V
```

426

ID(4)*ID(1).LT.ID(118)

YES

ID(4)=ID(4)+1

NO

426

NT=ID(118)/NW

ID(1)=NT/ID(4)

428

ID(4)*ID(1).LT.NT

YES

ID(1)=ID(1)+1

NO

429

ID(1)=ID(1)*NW

JCNT=JNEW

```
             :
             :
     431     V
 ****************************
/                            \
 CALL DRUM(ID(1),ISU(KNEW),JCNT
 ,0)
\                            /
 ****************************
             :
             V
            / \
           ( 235 )   442
            \ /

 / \
( 230 )------------------>:
 \ /
     433     V
   *******************
   *                 *
   *  ID(1)=KNT-KNEW *
   *                 *
   *******************
             :
             :
     434     V
            .*.
         .*'     '*.     YES
      .*'  ID(1).GT.0  '*.------------------------------->( 232 )   439
      '*.                .*'
         '*.          .*'
            '*'
             :  NO
             :
     435     V
      *************
      *           *
      *  ID(1)=Nr *
      *           *
      *************
             :
             :
             :
```

```
                         :
                         :
                 436     V
                ***************
               /               \
              /  DO 231 I=1,NW  \
              :                 :
              :  J=KNEW+I-1     :
              :                 :
              \  ISU(J)=0       /
               \               /
                ***************
  /   \                  :
 ( 232 )---------------->:
  \   /          439     V
              *********************
              *                   *
              *  ID(4)=1          *
              *                   *
              *  ID(118)=ID(1)    *
              *                   *
              *  JCNT=JNEW+ID(1)  *
              *                   *
              *********************
  /   \                  :
 ( 235 )---------------->:
  \   /          442     V
          *******************************
         /                               \
        /  CALL TAPEOU(LUN,ISTAT,ID(1),   \
        \  ISU(KNEW))                     /
         \                               /
          *******************************
                         :
                         :
                 443     V
                        .*.
                   .*'       '*.    YES                                     /   \
                .*'  ISTAT.GT.0  '*.------------------------------------>( 300 )    450
                '*.               .*'                                     \   /
                   '*.       .*'
                        '*'
  /   \                  :  NO
 ( 236 )---------------->:
  \   /                  :
                         :
                         :
                         :
                         V
```

```
                     :
                     :<----------------------- --
            444      V                          A
                    .*.                         :
                 .*'     '*.                    :
              .*'           '*.     YES         :
           .*'   JCNT.GE.JNT   '*.------------->:
           '*.                 .*'              : :
              '*.           .*'                 : :
                 '*.     .*'                    : :
                    '*'                         : :
                     : NO                       : :
            445      V                          : :
       *****************************           : :
      /                             \           : :
     /  CALL DRUM(ID(1),ISU(KNEW),JCNT \        : :
     :  ,0)                          :          : :
     :                               :          : :
     :  CALL DATAOU(LUN,ISTAT,ISU(KNEW :        : :
     \  ))                           /          : :
      \                             /           : :
       *****************************           : :
                     :                          : :
            447      V                          : :
                    .*.                         : :
  /   \        +  .*'   '*.       -,0           : :
 ( 300 )<-------.*'  ISTAT  '*.---------------->: :
  \   /         '*.       .*'                     :
   450             '*.  .*'                       :
                      '*'                         :
                                                  V
  /   \                                           
 ( 240 )------------------>:<----------------------
  \   /             448    V
             *************************
             *                       *
             *  NPING=NPING+ITHSEQ   *
             *                       *
             *************************
                           :
                           :
                           V
```

```
                           :
                           :
                     449   V
                          .*.
                       .*'   '*.
 / \      +         .*'         '*.    -,0
( 160 )<-------.*'    NPING-NMMAX    '*.------------------------------>( 100 )   109
 \   /           '*.                .*'                                  \   /
  30                '*.          .*'
                       '*.    .*'
                          '*'

 / \
( 300 )----------------->:
 \   /             450   V
                  **************************
                 /                        /
                /  PRINT 905,IEOF,ISTAT  /
               /                        /
              **************************
                         :
 / \                     :
( 400 )----------------->:
 \   /             453   V
                 *******************
                  '*.             .*'
                    '*.   END   .*'
                      '*.     .*'
                        '*. .*'
                          '*'
```

07 MAR 73   18:06:25.996

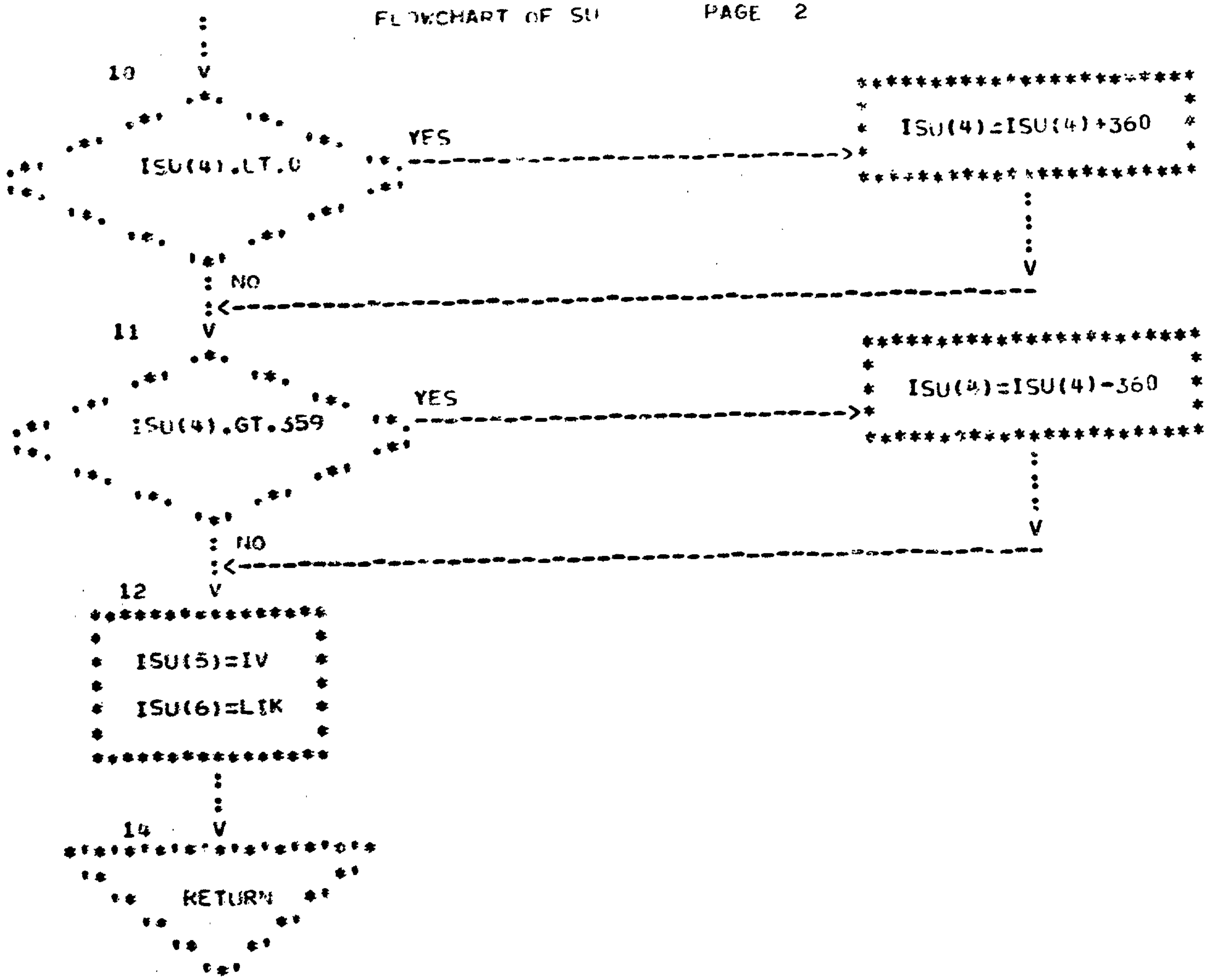
QCL FLO SU
TRACOR COMPUTING CORPORATION
FLOWCHART PROCESSOR LEVEL 1.0

```
C     SUBROUTINE SU ESTABLISHES A TARGET STATUS UNIT CONTAINING PRESENT
C     RANGE AND BEARING, PREDICTED RANGE AND BEARING, VARIANCE INDICA-
C     TOR AND LOG LIKELIHOOD RATIO.
      SUBROUTINE SU(ISU,IR,IB,LR,LB,IV,LIK)
      DIMENSION  ISU(1)
      ISU(1)=IR
      ISU(2)=IB
      ISU(3)=2*IR-LR
      ISU(4)=2*IB-LB
      IF(ISU(4).LT.0) ISU(4)=ISU(4)+360
      IF(ISU(4).GT.359)  ISU(4)=ISU(4)-360
      ISU(5)=IV
      ISU(6)=LIK
      RETURN
      END
```

0 DIAGNOSTIC MESSAGES.

```
              1
-------------------------------------
   SUBROUTINE SU ESTABLISHES A
   TARGET STATUS UNIT CONTAINING
   PRESENT

   RANGE AND BEARING, PREDICTED
   RANGE AND BEARING, VARIANCE
   INDICA-

   TOR AND LOG LIKELIHOOD RATIO.
-------------------------------------


              4
   ****************************
  /                            \
 (  SUBROUTINE                  )
 (  SU(ISU,IR,IB,LR,LB,IV,LIK)  )
  \                            /
   ****************************
                 :
                 :
              6  V
       ********************
       *                  *
       *  ISU(1)=IR       *
       *                  *
       *  ISU(2)=IB       *
       *                  *
       *  ISU(3)=2*IR-LR  *
       *                  *
       *  ISU(4)=2*IB-LB  *
       *                  *
       ********************
                :
                :
                :
                V
```

FLOWCHART OF SU PAGE 2



GCL FLG ICOMP
TRACOR COMPUTING CORPORATION
FLOWCHART PROCESSOR LEVEL 1.0

07 MAR 73   18:06:26.397

```
C     FUNCTION ICOMP FINDS OUT IF A GIVEN POSITION IS WITHIN A TRACKING
C     WINDOW.
      FUNCTION ICOMP(IR,IB,LLR,LLB,LUR,LUB)
      IF(IR.LT.LLR)GO TO 50
      IF(IR.GT.LUR) GO TO 100
      ICOMP=IB-LLB
      IF(ICOMP.GT.180) ICOMP=ICOMP-360
      IF(ICOMP.LT.-180) ICOMP=ICOMP+360
      IF(ICOMP.LT.0) GO TO 60
      ICOMP=IB-LUB
      IF(ICOMP.GT. 180)  ICOMP=ICOMP-360
      IF(ICOMP.LT.-180)  ICOMP=ICOMP+360
      IF(ICOMP.GT.0)  GO TO 110
   40 ICOMP=0
      GO TO 200
   50 ICOMP=-2
      GO TO 200
   60 ICOMP=-1
      GO TO 200
  100 ICOMP=2
      GO TO 200
  110 ICOMP=1
  200 RETURN
      END
```

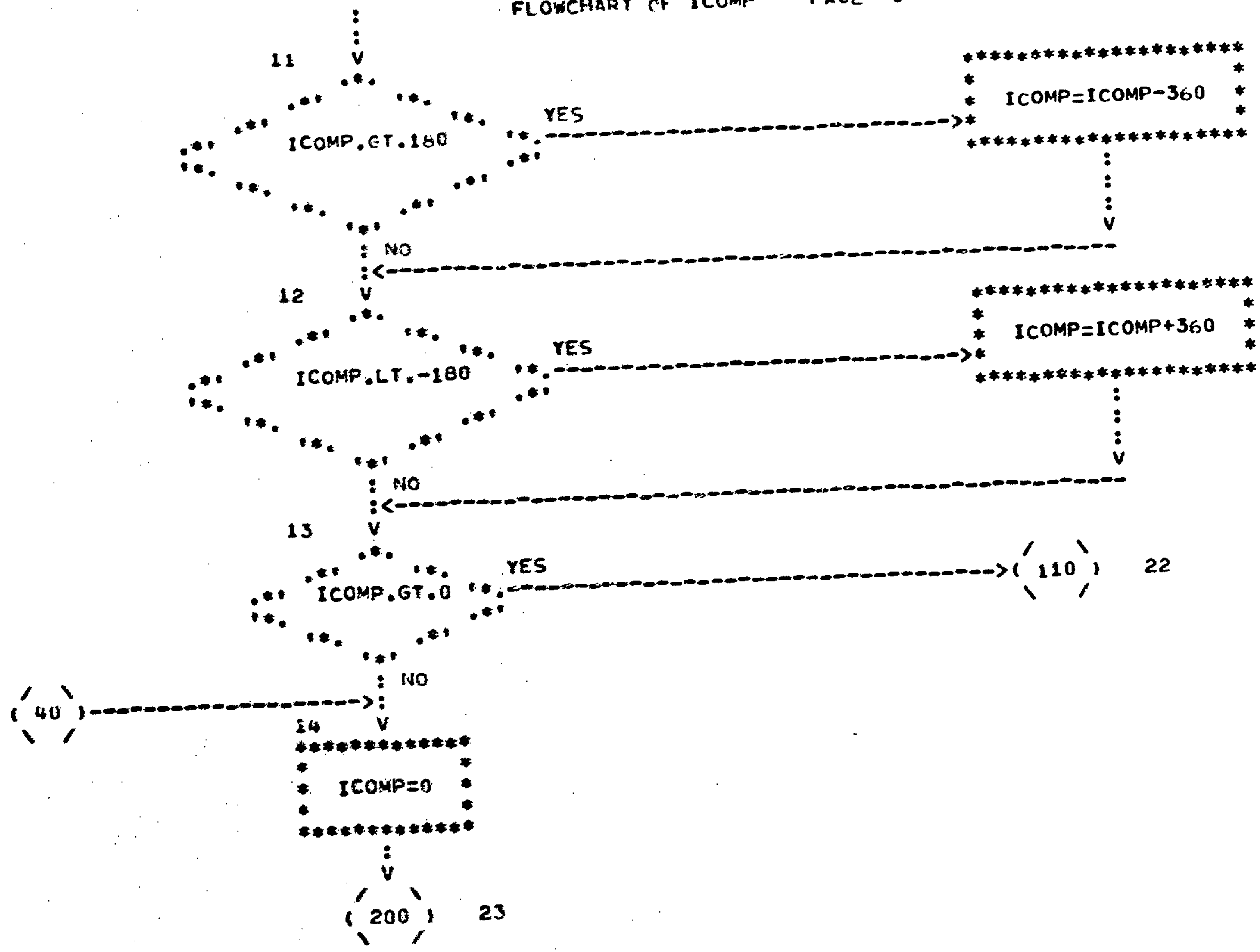
0 DIAGNOSTIC MESSAGES.

```
                1
 ----------------------------------
   FUNCTION ICOMP FI?DS OUT IF A
   GIVEN POSITION IS WITHIN A
   TRACKING

   WINDOW.
 ----------------------------------

                3
   ****************************
  /                            \
 (  FUNCTION                    )
 (  ICOMP(IR,IB,LLR,LLB,LUR,LUB) )
  \                            /
   ****************************
                :
                :
         4      V
               .*.
           .*'     '*.     YES
       .*'   IR.LT.LLR  '*.------------------------------>( 50 )    16
       '*.              .*'
           '*.      .*'
               '*'
                : NO
                :
         5      V
               .*.
           .*'     '*.     YES
       .*'   IR.GT.LUR  '*.------------------------------>( 100 )   20
       '*.              .*'
           '*.      .*'
               '*'
                : NO
                :
         6      V
       *****************
       *               *
       *  ICOMP=IB-LLB *
       *               *
       *****************
                :
                :
                :
                :
                V
```

FLOWCHART OF ICOMP   PAGE 2

```
                 :
                 :
        7        V
                .*.
             .*'   '*.
          .*'         '*.     YES                               *******************
       .*'   ICOMP.GT.180  '*.-------------------------------->*                 *
       '*.                 .*'                                 *  ICOMP=ICOMP-360 *
          '*.         .*'                                      *                 *
             '*.   .*'                                         *******************
                '*'                                                     :
                 : NO                                                   :
                 :<-----------------------------------------------------V
        8        V
                .*.
             .*'   '*.
          .*'         '*.     YES                               *******************
       .*'   ICOMP.LT.-180 '*.-------------------------------->*                 *
       '*.                 .*'                                 *  ICOMP=ICOMP+360 *
          '*.         .*'                                      *                 *
             '*.   .*'                                         *******************
                '*'                                                     :
                 : NO                                                   :
                 :<-----------------------------------------------------V
        9        V
                .*.
             .*'   '*.      YES
          .*' ICOMP.LT.0 '*.---------------------------------->( 60 )   18
          '*.            .*'
             '*.      .*'
                '*'
                 : NO
                 :
        10       V
       *****************
       *               *
       *  ICOMP=IB-LUB *
       *               *
       *****************
                 :
                 :
                 :
                 V
```

FLOWCHART OF ICOMP PAGE 3



```
  / \
( 50 )------------------->:
  \ /                     
                     16   V
                   **************
                   *            *
                   *  ICOMP=-2  *
                   *            *
                   **************
                          :
                          V
                        /   \
                      ( 200 )    23
                        \   /

  / \
( 60 )------------------->:
  \ /
                     18   V
                   **************
                   *            *
                   *  ICOMP=-1  *
                   *            *
                   **************
                          :
                          V
                        /   \
                      ( 200 )    23
                        \   /

  / \
( 100 )------------------>:
  \ /
                     20   V
                   *************
                   *           *
                   *  ICOMP=2  *
                   *           *
                   *************
                          :
                          V
                        /   \
                      ( 200 )    23
                        \   /
```

```
 / \
( 110 )------------------->:
 \ /                22    V
                    ************
                    *          *
                    *  ICOMP=1 *
                    *          *
                    ************
 / \                       :
( 200 )------------------->:
 \ /                23    V
                 ****************
                  **          **
                   **  RETURN **
                    **      **
                     **    **
                       ***
```

QCL FLO IORD
TRACOR COMPUTING CORPORATION
FLOWCHART PROCESSOR LEVEL 1.0

07 MAR 73   18:06:27.033

```
C      FUNCTION IORD FINDS OUT IF A GIVEN POSITION VECTOR IS LESS THAN
C      OR GREATER THAN A GIVEN POSITION VECTOR.
       FUNCTION IORD(IR,IB,LR,LB)
       IF(IR-LR) 50,10,100
    10 IF(IB-LB)  60,40,110
    40 IORD=0
       GO TO 200
    50 IORD=-2
       GO TO 200
    60 IORD=-1
       GO TO 200
   100 IORD=2
       GO TO 200
   110 IORD=1
   200 RETURN
       END
```

0 DIAGNOSTIC MESSAGES.

```
                         1
          ----------------------------------
            FUNCTION IORD FINDS OUT IF A
            GIVEN POSITION VECTOR IS LESS
            THAN

            OR GREATER THAN A GIVEN
            POSITION VECTOR.
          ----------------------------------

                         3
            ***************************
           /                           \
          (  FUNCTION IORD(IR,IB,LR,LB)  )
           \                           /
            ***************************
                         :
                         :
                   4     V
                        .*.
          +         .*'     '*.     -
 ( 100 )<--------.*'   IR-LR   '*.---------------------------->( 50 )   8
    12            '*.           .*'
                      '*.   .*'
                         '*'
                         :  0
 ( 10 )----------------->:
                   5     V
                        .*.
          +         .*'     '*.     -
 ( 110 )<--------.*'   IB-LB   '*.---------------------------->( 60 )   10
    14            '*.           .*'
                      '*.   .*'
                         '*'
                         :  0
 ( 40 )----------------->:
                   6     V
                 ************
                 *          *
                 *  IORD=0  *
                 *          *
                 ************
                       :
                       :
                       V
```

```
                              :
                              V
                             / \
                            ( 200 )    15
                             \ /

  / \
 ( 50 )------------------->:
  \ /                  8    V
                       *************
                       *           *
                       *  IORD=-2  *
                       *           *
                       *************
                              :
                              V
                             / \
                            ( 200 )    15
                             \ /

  / \
 ( 60 )------------------->:
  \ /                  10   V
                       *************
                       *           *
                       *  IORD=-1  *
                       *           *
                       *************
                              :
                              V
                             / \
                            ( 200 )    15
                             \ /

  / \
 ( 100 )------------------>:
  \ /                  12   V
                       *************
                       *           *
                       *  IORD=2   *
                       *           *
                       *************
                              :
                              V
                             / \
                            ( 200 )    15
                             \ /
```

```
 / \
( 110 )------------------->:
 \ /                14     V
                    ************
                    *          *
                    *  IORD=1  *
                    *          *
                    ************
 / \                       :
( 200 )------------------->:
 \ /                15     V
                ********************
                 **              **
                  **   RETURN   **
                   **          **
                     **      **
                        ***
```

QCL FLO DRUM
TRACOR COMPUTING CORPORATION
FLOWCHART PROCESSOR LEVEL 1.0

07 MAR 73   18:06:27.434

```
      SUBROUTINE DRUM(N,IB,IADD,ISW)
      DIMENSION IB(1),IS(64)
      NLEN=64
      ISEC=(IADD-1)/NLEN+1
      NT=ISEC*NLEN+1-IADD
      IF(NT.NE.NLEN) GO TO 10
      ISEC=ISEC-1
      NT=0
   10 IF(N.LT.NT) GO TO 20
      IF(ISW.EQ.0) GO TO 15
      CALL STORE(ISEC+1,IB(NT+1),N-NT,5)
      GO TO 19
   15 CALL FETCH(ISEC+1,IB(NT+1),N-NT,5)
   19 IF(NT.EQ.0) GO TO 40
   20 CALL FETCH(ISEC,IS,NLEN,5)
      J=NLEN-NT
      IF(ISW.EQ.0) GO TO 30
      DO 25 I=1,NT
      J=J+1
   25 IS(J)=IB(I)
      CALL STORE(ISEC,IS,NLEN,5)
      GO TO 40
   30 DO 35 I=1,NT
      J=J+1
   35 IB(I)=IS(J)
   40 IADD=IADD+N
      RETURN
      END
```

0 DIAGNOSTIC MESSAGES.

```
                 1
     *******************************
    /                               \
   (  SUBROUTINE DRUM(N,IB,IADD,IS) )
    \                               /
     *******************************
                    :
                    :
             3      V
       *************************
       *                       *
       *  NLEN=64              *
       *                       *
       *  ISEC=(IADD-1)/NLEN+1 *
       *                       *
       *  NT=ISEC*NLEN+1-IADD  *
       *                       *
       *************************
                    :
                    :
             6      V
                   .*.
               .*'     '*.        YES
           .*'  NT.NE.NLEN  '*.------------------>:
           '*.              .*'                   :
               '*.     .*'                        :
                   '*'                            :
                    : NO                          :
                    :                             :
             7      V                             :
           ****************                       :
           *              *                       :
           *  ISEC=ISEC-1 *                       :
           *              *                       :
           *  NT=0        *                       :
           *              *                       :
           ****************                       :
                    :                             V
  / \               :                             :
 ( 10 )------------>:<----------------------------
  \ /               :
                    :
                    :
                    :
                    :
                    V
```

```
                    :
                    :
          9         V
                   .*.
               .*'     '*.      YES                                  /  \
           .*'   N.LT.NT   '*.------------------------------------->( 20 )   15
           '*.             .*'                                       \  /
               '*.     .*'
                   '*'
                    : NO
                    :
          10        V
                   .*.
               .*'     '*.      YES
           .*'   ISW.EQ.0  '*.----------------->:
           '*.             .*'                  :
               '*.     .*'                      :
                   '*'                          :
                    : NO                        :
                    :                           :
          11        V                           :
      ******************************            :
     /                              \           :
    /  CALL STORE(ISEC+1,IB(NT+1),N- \          :
    \  NT,5)                         /          :
     \                              /           :
      ******************************            :
                    :                           :
                    :                           :
                    V                           :
                    ---------------------------->:
                                                :  :
   /  \                                         V  :
  ( 15 )---------------->:<---------------------   :
   \  /                  V                         :
          13        V                              :
      ******************************               :
     /                              \              :
    /  CALL FETCH(ISEC+1,IB(NT+1),N- \             :
    \  NT,5)                         /             :
     \                              /              :
      ******************************               :
   /  \                 :                          V
  ( 19 )---------------->:<-------------------------
   \  /                 :
                        :
                        :
                        V
```

```
                    :
                    :
               14   V
                   .*.
               .*'     '*.     YES
            .*'  NT.EQ.0  '*.------------------------------>( 40 )   26
            '*.           .*'
               '*.     .*'
                   '*'
                    : NO
   ( 28 )---------->:
               15   V
       ****************************
      /                            \
     /  CALL FETCH(ISEC,IS,NLEN,5)  \
     \                              /
      \                            /
       ****************************
                    :
                    :
               16   V
             ***************
             *             *
             *  J=NLEN-NT  *
             *             *
             ***************
                    :
                    :
               17   V
                   .*.
               .*'     '*.     YES
            .*'  ISW.EQ.0 '*.------------------------------>( 30 )   23
            '*.           .*'
               '*.     .*'
                   '*'
                    : NO
               18   V
             **************
            /              \
           /  DO 25 I=1,NT  \
           :                :
           :  J=J+1         :
           :                :
           \  IS(J)=IH(I)   /
            \              /
             **************
                    :
                    :
                    V
```

```
                         :
                         :
                  21     V
           ****************************
          /                            \
         /  CALL STORE(ISEC,IS,NLEN,S)   \
         \                              /
          \                            /
           ****************************
                         :
                         :
                         V
                         ------------------------>:
                                                  :
                                                  :
  / \                                             :
 ( 30 )------------------>:                       :
  \ /             23      V                       :
                ***************                   :
               /               \                  :
              /  DO 35 I=1,NT   \                 :
              :                 :                 :
              :  J=J+1          :                 :
              :                 :                 :
              \  IB(I)=IS(J)    /                 :
               \               /                  :
                ***************                   :
                          :                       V
  / \                     :                       :
 ( 40 )------------------>:<-----------------------
  \ /             26      V
               *****************
               *               *
               *  IADD=IADD+N  *
               *               *
               *****************
                          :
                          :
                  27      V
             *'*'*'*'*'*'*'*'*'*'*
              '*                *'
               '*   RETURN     *'
                '*            *'
                 '*         *'
                  '*      *'
                    '*'
```

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Tracor, Inc. 6500 Tracor Lane Austin, TX 78721		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP -	
3. REPORT TITLE COMPUTER AIDED PROCESSING FOR ACTIVE AND PASSIVE SONAR SYSTEMS			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) FINAL REPORT			
5. AUTHOR(S) (First name, middle initial, last name) H. A. Reeder and D. W. Hamm			
6. REPORT DATE 14 February 1973	7a. TOTAL NO. OF PAGES 238		7b. NO. OF REFS 6
8a. CONTRACT OR GRANT NO. N00024-70-C-1274 b. PROJECT NO. c. 035 001 d.	9a. ORIGINATOR'S REPORT NUMBER(S) T73-AU-9519-U 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) None		
10. DISTRIBUTION STATEMENT			
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Commander Naval Ship Systems Command Code 302-4		

13. ABSTRACT

A general framework for a computer system that accepts and analyzes the vast quantity of data generated by a modern sonar suite has been developed. The output of this computer system is an array of alerting functions that measure the likelihood that a given coordinate vector is the location of a target. In particular, the framework combines the output of active high- and low-Doppler sonar processors and wideband and narrowband passive processors. The active high- and low-Doppler processor portion was developed and tested during this study. Performance tests using simulated data established that the combined active processor gave better performance than each individual processor and its single output channel gave more uniform performance over variations in target Doppler than was available with the two separate channels. An observer test was conducted using ARL processed AN/SQS-23 recorded sea data with injected target signals.

DD FORM 1 NOV 65 1473

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computer Aided Detection						
Data Processing						
Computer Programs						
Statistical Tests						